Computational Analysis and Interpretation of Prokaryotic High-throughput Expression Data

Von der Fakultät für Lebenswissenschaften

der Technischen Universität Carolo-Wilhelmina

zu Braunschweig

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

genehmigte

D i s s e r t a t i o n

von Maurice Patrick Scheer
aus Berlin

1. Referent: Prof. Dr. Michael Steinert
2. Referent: Prof. Dr. Frank Klawonn
eingereicht am: 25.06.2008
mündliche Prüfung (Disputation) am: 11.09.2008

Druckjahr 2008

Nina und Bruno

Vorveröffentlichungen der Dissertation:
Teile dieser Arbeit wurden mit Genehmigung der Fakultät für Lebenswissenschaften, vertreten durch den Mentor der Arbeit, in folgenden Beiträgen vorab veröffentlicht:

Publikationen:

Hiller, K., Grote, A., Scheer, M., Münch, R. & Jahn, D. (2004) PrediSi: prediction of signal peptides and their cleavage positions. *Nucleic Acids Res.* **32**, W375–W379.

Grote, A., Hiller, K., Scheer, M., Münch, R., Nörtemann, B., Hempel, D.C. & Jahn, D. (2005) JCat: a novel tool to adapt codon usage of a target gene to its potential expression host. *Nucleic Acids Res.* **33**, W526–W531.

Münch, R., Hiller, K., Grote, A., Scheer, M., Klein, J., Schobert, M. & Jahn, D. (2005) Virtual Footprint and PRODORIC: an integrative framework for regulon prediction in prokaryotes. *Bioinformatics* **21**, 4187–4189.

Scheer, M., Klawonn, F., Münch, R., Grote, A., Hiller, K., Choi, C., Koch, I., Schobert, M., Härtig, E., Klages, U. & Jahn, D. (2006) JProGO: a novel tool for the functional interpretation of prokaryotic microarray data using Gene Ontology information. *Nucleic Acids Res.* **34**, W510–W515.

Choi, C., Münch, R., Leupold, S., Klein, J., Siegel, I., Thielen, B., Benkert, B., Kucklick, M., Schobert, M., Barthelmes, J., Ebeling, C., Haddad, I., Scheer, M., Grote, A., Hiller, K., Bunk, B., Schreiber, K., Retter, I., Schomburg, D. & Jahn, D. (2007) SYSTOMONAS - an integrated database for systems biology analysis of *Pseudomonas*. *Nucleic Acids Res.* **35**, D533–D537.

Eingereichte Publikationen und Konzepte:

Schreiber, K., Scheer, M., Garbe, J., Hiller, K., Benkert, B., Bös, N., Thielen, B., Schomburg, D., Brors, B., Buer, J., Jahn, D. & Schobert, M. Role of the universal stress protein K (UspK) in the transcriptional and metabolic adaptation of *Pseudomonas aeruginosa* to anaerobic survival via pyruvate fermentation. Manuscript concept in preparation

Benkert, B., Schreiber, K., Scheer, M., Geffers, R., Jahn, D. & Schobert, M. The regulon of the nitrate response regulator NarL from *Pseudomonas aeruginosa*. Manuscript concept in preparation

Tagungsbeiträge:

Scheer, M., Klawonn, F., Münch, R., Schobert, M., Härtig, E., Grote, A., Hiller, K., Koch, I., Klages, U. & Jahn, D. (2004) Interpretation of Bacterial Microarray Data Using Automated Numerical Evaluation of Gene Ontology Information – The Influence of Oxygene Tension on Bacterial Gene Expression. (Poster) *German Conference on Bioinformatics (GCB) 2004*, Bielefeld

Scheer, M., Klawonn, F., Münch, R., Grote, A., Hiller, K., Koch, I., Klages, U. & Jahn, D. (2004) Functional Analysis of Bacterial Gene Expression Data Based on Automated Numerical Evaluation of Gene Ontology Annotation: How Oxygen Influences Bacterial Gene Expression. (Poster) *International Conference on Systems Biology (ICSB) 2004*, Heidelberg

Scheer, M., Klawonn, F., Münch, R., Grote, A., Hiller, K., Koch, I., Klages, U. & Jahn, D. (2005) A Tool for Threshold Independent Functional Interpretation of Prokaryotic Microarray using Gene Ontology – Application to Microarray Data from *E. coli*. (Poster) *German Conference on Bioinformatics (GCB) 2005*, Hamburg

Contents

Zusammenfassung

DNA-Mikroarray-basierte Transcriptomics-Experimente liefern große Mengen wertvoller Informationen über die transkriptionelle Aktivität sämtlicher Gene eines Mikroorganismus. Nach der Präprozessierung der erhaltenen Rohdaten erfolgt normalerweise die funktionelle Interpretation. Dies manuell durchzuführen, ist sehr zeitintensiv und ein Überblick über die relevanten Funktionen lässt sich so schwer gewinnen. Daher wurde in der vorliegenden Arbeit eine neue integrative Software-Suite für die funktionelle Auswertung von Genexpressionsdaten (JProGO) entwickelt, welche – basierend auf der Gene Ontology (GO) als Klassifikationssystem – diejenigen biologischen Funktionen und Prozesse identifiziert, deren Expressionsprofile sich zwischen den beiden untersuchten Bedingungen signifikant unterscheiden. Die Software unterstützt mehr als 20 verschiedene prokaryotische Spezies. Neben dem in der Literatur häufig für eine funktionelle Interpretation benutzten Schwellenwert-basierten exakten Fisher-Test und dem Schwellenwert-freien t-, Kolmogorov-Smirnov (KS)- sowie Mann-Whitney U-Test bietet die Software-Suite geeignete Korrekturmethoden für das multiple Testen an: die Bonferroni-Korrektur und die False Discovery Rate-Methode. Weitere Funktionalitäten umfassen die Erkennung von alternativen Gennamen, die Unterstützung verschiedener Expressionsdaten-Typen und die Visualisierung der berechneten Ergebnisse als Tabelle und als Untergraph von GO, welcher die azyklische Graphenstruktur berücksichtigt. Das Programm wurde mit Expressionsdaten der klassischen bakteriellen Modellorganismen, *Escherichia coli* und *Bacillus subtilis*, evaluiert. Hierbei wurden der Einfluß und die Willkür des Schwellenwerts des exakten Fisher-Tests genauer untersucht. Danach wurden in einer vergleichenden Fallstudie die Schwellenwert-freien Methoden mit ausgewählten Expressionsdatensätzen von *E. coli* evaluiert. Dabei erwies sich der U-Test als gute Alternative zum KS- und t-Test, falls die Zahl gleicher Ränge nicht zu gross ist. Außerderm wurde der Einfluß des Expressionsdaten-Types, Expressionsquotienten und Teststatistiken (p-Werte), untersucht, wobei der Einsatz von Teststatistiken empfohlen wird, falls genügend Replikate vorliegen. Ein direkter Vergleich der Analyse-Ergebnisse von Schwellenwert-basierten (Fisher-Test) mit Schwellenwert-freien (U-Test) Algorithmen bestätigte die erwartete schwache Korrelation bezogen auf die p-Werte aller GO-Terme. Zugleich ergab sich aber interessanter Weise eine große Überlappung bezüglich der signifikanten GO-Knoten. Nach den Fallstudien, in denen JProGO mit präprozessierten Expressionsdaten verwendet wurde, wurden im Institut gewonnene Rohexpressiondaten des medizinisch bedeutsamen Bakteriums *Pseudomonas aeruginosa* ausgewertet. Es wurde eine kombinierte Low- und Mid-Level-Analyse mit Bioconductor durchgeführt, und die errechneten Expressionswerte wurden dann funktionell analysiert. Hierbei wurde der Einfluß verschiedener Präprozessierungsalgorithmen auf das Ergebnis von JProGO-gestützten High-Level-Analysen untersucht. Zudem wurden einige signifikante GO-Knoten identifiziert, welche mit der Erwartung an das Experiment übereinstimmen. Diese umfassen beim Vergleich von anaerob mit und ohne Nitrat kultivierten Wildtyp-PAO1-Zellen u.a. die GO-Terme Zitronensäure-Zyklus, aerobe Atmung und Nitrat-Reduktase-Aktivität. Schließlich wurde die funktionelle Analyse, welche bei JProGO auf GO-Terme beschränkt war, auf eine weitere biologische Gen-Gruppierung, das Regulon, erweitert. Hierfür wurden experimentell validierte Regulons der PRODORIC-Datenbank eingesetzt. Ein Prototyp dieses neuen Programms wurde mit geeigneten Datensätzen von *E. coli*-Stämmen, in denen je ein Transkriptionsfaktor ausgeschaltet war, evaluiert. Die Ergebnisse entsprachen der Erwartung gut. Der KS-Test schnitt dabei am besten ab, dicht gefolgt vom U-Test.

Summary

DNA microarray-based transcriptomics experiments provide large amounts of valuable data on the transcriptional activity of all genes of a single microorganism at once. After performing the obligatory preprocessing of the obtained raw data, normally the functional interpretation follows. Performing this manually is a tedious, very time-consuming task and it is difficult to obtain a comprehensive overview on the most relevant functions this way. Therefore, in the thesis at hand an integrative novel program suite for the functional interpretation of microarray gene expression data (JProGO) was developed which – based on the Gene Ontology (GO) classification system – identifies those biological functions and processes that significantly differ in their expression profiles when comparing two experimental conditions. The software supports a broad range of more than 20 prokaryotic species. Amongst offering the cut-off based Fisher's exact test as well as the cut-off free Student's t-test, Kolmogorov-Smirnov (KS) test and unpaired Wilcoxon test (U-test), which were commonly described in the literature for similar purposes, appropriate methods of correcting the multiple testing effect are provided by JProGO: the Bonferroni and the False Discovery Rate method. Further features of the program are the recognition of alternative gene names, support of different types of expression data, and the visualization of the obtained results as both, a tabular view and a subgraph of GO which considers its directed acyclic graph structure. The tool was tested with expression data from the classical bacterial model organisms *Escherichia coli* and *Bacillus subtilis*. In this context, the influence and arbitrariness of the threshold value for the cut-off based Fisher's exact test was elucidated. Subsequently, in a comparative case study the cut-off free methods were evaluated on selected expression data sets from *E. coli* and the U-test was found to be a good alternative to the Kolmogorov-Smirnov test and Student's t-test, if the number of equal ranks is not too high. Furthermore, the influence of the type of expression data – expression ratios and p-values – was investigated emphasizing the use of test statistics when a sufficient number of replicates is available. A direct comparison of the analysis results of threshold-based (Fisher's exact test) to threshold-free (U-test) tests confirmed the expected weak correlation between the p-values over all GO nodes, but interestingly revealed a high partial overlap among the significant nodes. After the case studies which used JProGO with preprocessed prokaryotic expression data sets, in-house raw expression data from the medically relevant pathogen *Pseudomonas aeruginosa* were analyzed. A combined low-level and mid-level analysis using Bioconductor was performed and the computed expression levels were interpreted in a high-level functional analysis. In this context, the impact of different preprocessing algorithms on the outcome of the JProGO-based high-level analysis was investigated. Several significant GO nodes, which fit with the expectation on the experiment, were identified. They comprise, for example, the GO terms tricarboxylic acid cycle, aerobic respiration and nitrate reductase activity for the comparison of wild type PAO1 cells grown anaerobically with and without nitrate. Finally, the functional analysis, which was restricted to GO terms in JProGO, was expanded towards another biological grouping of genes, the regulon. For this purpose, experimentally validated regulons of the PRODORIC database were utilized. A prototype of this new tool was evaluated comprehensively with appropriate expression data sets from *E. coli* strains in which in each case a transcriptional regulator was knocked out. The obtained results are in good agreement with the clear expectation on the affected regulons. The KS-test performed best, whereas the U-test was almost as good.

1 Introduction

1.1 High-throughput Technologies in Biosciences and Application of Bioinformatics

Due to the development of new technologies in molecular biology in the recent years, the amount of biological data increased dramatically. The introduction of novel DNA sequencing techniques caused an exponential growth of DNA sequence and whole genome data since the early 1980s (Kanehisa and Bork, 2003). In order to allow for a structured storage and update of this large bulk of data as well as for a fast and targeted access to individual data sets, conventional methods such as publication in journal articles or storage in text files of differing formats (flat files) were not qualified for. For these purposes database management systems are well suited. The above mentioned amount of sequence data was stored in publicly accessible sequence databases, which is regarded as the birth of bioinformatics (Hocquette, 2005). More in-depth information on the storage of DNA sequence, deduced amino acid sequence and other biological data as well as their bioinformatical representation can be found in chapter 1.3. Besides the field of genomics and valuable DNA sequence information, three other areas of research based on high-throughput technologies have evolved. They are successively build up on genomics (Singh and Nagaraj, 2006) and are, therefore, sometimes also referred to as functional genomics (Hocquette, 2005). They represent the information flow in the cell comprising DNA, RNA, proteins and enzyme-catalyzed metabolism:

1. Transcriptomics: genome-wide measurement of gene expression
2. Proteomics: analysis of (nearly) all proteins encoded by one genome
3. Metabolomics: analysis of a cell's metabolites

A focus of this work is on the bioinformatical analysis of data from transcriptomics (see below).

1.2 DNA Microarrays for High-throughput Gene Expression Profiling and Transcriptomics

1.2.1 Definition, Benefits and Relevance

The invention of DNA chips in 1995 – also known as DNA microarrays, biochips and gene chips – had a great impact on biological and biomedical research, especially on the field of gene expression analysis (see Schena *et al.*, 1995; Chee *et al.*, 1996; Schena, 2003; Chaudhuri, 2005). While previous techniques for studying gene expression like Northern blot hybridization and RT-PCR can only be conducted with one gene at a time, miniaturized DNA microarrays allow to measure the expression of thousands to hundred thousands of genes in parallel, in a single experiment (Hardiman, 2004). Thus, applied to microorganisms, DNA microarrays constitute a valuable high-throughput technology, which even enables to study the expression profile of all genes of a genome. Because of this fundamental advantage, nowadays, most of the gene expression data are derived from microarrays. This is also reflected by the steadily growing number of publications

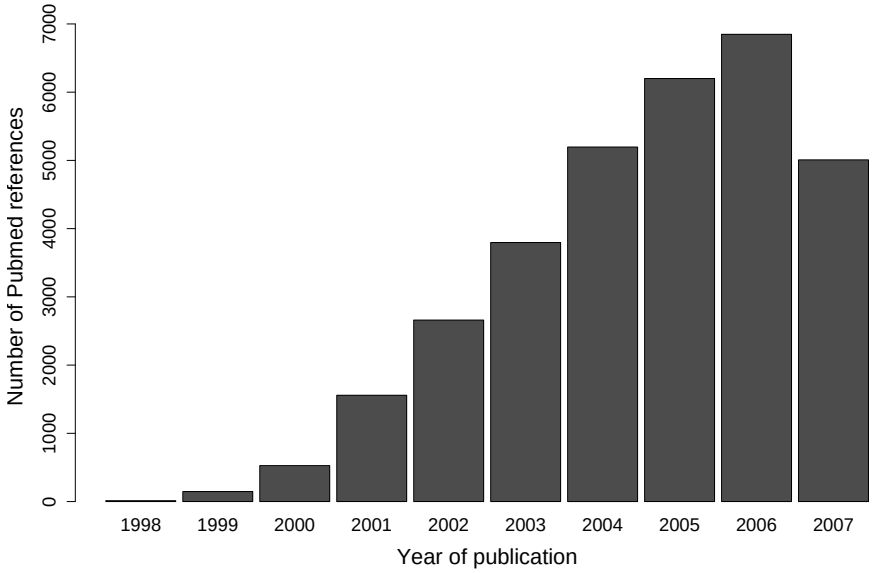


Figure 1: The growing annual number of microarray-related publications obtained from the Pubmed database since 1998. In a search with the Pubmed web interface the medical subject headings "Oligonucleotide Array Sequence Analysis" and "Microarray Analysis" were used. With the exception of 2007, in all other years the number of publications increased.

per year using this technology (Fig. 1, see also Chaudhuri, 2005). As a consequence, for many sequenced organisms huge amounts of high-throughput expression data are available, from which useful insights might be obtained with the help of appropriate bioinformatics analyses (see e.g. chapter 1.4.3).

Besides the above mentioned clear advantages of the microarray technology, a slight drawback is the experimental noise often inherent in its measurements. It can be at least partially compensated by performing a sufficient number of replicate experiments (see Knudsen, 2002). In addition, during the obligatory preprocessing of the raw data obtained from any microarray experiment (chapter 1.4.1) the noise can be partially reduced with the help of statistical models which are applied to the data (Gentleman *et al.*, 2005). Furthermore, well tried low-throughput methods such as quantitative RT-PCR (see above) can be used for the validation of the microarray expression data or for a more precise quantitation of the expression levels of selected genes.

Before going into the details of the application of DNA microarrays in large-scale gene expression profiling, it should be mentioned that the area of application of microarrays is not restricted to determining the expression levels of genes. DNA microarrays have also been successfully applied to genotyping, e.g. detecting single nucleotide polymorphisms, and identifying transcription factor binding sites. Protein microarrays were utilized to reveal protein-protein interactions or to identify the substrates of protein kinases or the target proteins of small organic molecules (Knudsen, 2002; Venkatasubbarao, 2004; Chaudhuri, 2005; Aguilar-Mahecha *et al.*, 2006; Kreutzberger, 2006).

1.2.2 Functionality of the Technology and Used Platforms

Like with Northern blotting, Southern blotting and PCR, the DNA microarray technology uses physical properties characteristic for single-stranded DNA molecules. The pairing of complementary bases is exploited which is mediated by hydrogen bonds between adenine and thymine (2 hydrogen bonds) as well as cytosine and guanine (3 hydrogen bonds). It leads to the hybridization of complementary DNA strands to form double-stranded stretches of DNA (Alberts *et al.*, 1994 and Gentleman *et al.*, 2005). At distinct fixed positions on the surface of a DNA microarray, which normally is based on a rectangular slide made up of glass or polymer, single-stranded DNA molecules have been chemically attached. They are called spots or features – sometimes they are also designated as probes – and form a regular lattice on the array. Thousands of such spots, which differ in the sequence of their DNA fragment, can be present on a microarray slide and each spot contains millions of identical copies of the respective DNA molecule (Causton *et al.*, 2003). Depending on the used microarray platform (see below) the spots of a gene expression array either represent complete or partial cDNAs of the genes of a certain organism, for example all genes of *Escherichia coli* K-12. The DNA spots on the array can hybridize to single-stranded cDNA molecules obtained from reverse transcription of RNA isolated from a sample of interest, e.g. the RNA molecules of anaerobically grown *E. coli* K-12 cells. Thus, they allow a semi-quantitative detection of the expression levels of the corresponding genes. For this purpose, the nucleic acids of the sample are labeled with a fluorescent dye during reverse transcription. Bound fluorescent cDNA probes are detected by a scanner of the microarray measurement setup (Knudsen, 2002).

While the principal functionality described above is the same for all DNA microarrays, several different microarray platforms exist. They mainly differ in the length of the DNA fragments and how these were attached to the surface of the array. According to these criteria two broad groups of platforms can be distinguished (Hardiman, 2004; Chaudhuri, 2005; Gentleman *et al.*, 2005):

1. full-length cDNA arrays (cDNA arrays)
2. high-density oligonucleotide arrays (oligonucleotide arrays)

cDNA arrays:
As their name suggests, the spots of full-length cDNA arrays either consist of the complete cDNA molecules or of large fragments (> 60 nucleotides) that represent some or all transcripts of the organism under investigation. Thus, the length of the DNA fragments in the spots ranges from several dozens to thousands of nucleotides. The number of the spots corresponds to the number of all genes of the organism or the selected subset of genes of special interest, e.g. specific for a certain tissue in the case of higher eukaryotes. The cDNA features are normally spotted or contact-printed onto the surface of the microarray with the help of a pin-based robotic arrayer or a device that is able to dispense tiny drops of liquid, also called inkjet microdispensing liquid handling system (Hardiman, 2004). The DNA fragments are obtained from cDNA clone libraries and the solid slide used for attaching them is often made up of derivative glass. Normally, two differently labeled sample cDNA populations derived from two biological sources – e.g. from two different cultivation conditions, labeled with red and green fluorescent dye – are hybridized on one cDNA array. Hence these arrays are also called two-color spotted microarrays (Gentleman *et al.*, 2005). Full-length cDNA arrays are either produced in

academic research laboratories themselves or can be obtained ready for use from commercial vendors such as Incyte and Agilent. One critical point to consider with full-length cDNA arrays are the different hybridization temperatures of the individual genes.

Oligonucleotide arrays:
The spots of high-density oligonucleotide arrays contain DNA fragments ranging in length between about 15 to 70 nucleotides (Chaudhuri, 2005). The sequence composition of these DNA molecules is designed *in silico*, whereas each oligonucleotide corresponds to one part of a gene's cDNA sequence. Vice versa each gene is normally represented by several different oligonucleotides and, thus, by distinct spots on the array, which is also referred to as probe redundancy (Hardiman, 2004; Chaudhuri, 2005). The most widespread oligonucleotide array platform is GeneChip® produced by Affymetrix (Lipshutz *et al.*, 1999) which pioneered this field (Hardiman, 2004) and offers ready-to-use DNA chips. Here, light-directed DNA synthesis, a combination of photolithography and solid-phase DNA synthesis, is used to polymerize the different oligos directly on the chip at the desired positions (Hardiman, 2004). For each oligonucleotide that perfectly matches a small part of a gene's cDNA (perfect match) another sequence exists that differs from the former in the base at the middle position (mismatch). Both oligos together, perfect match and mismatch, are designated as probe pair and each gene is represented by 16 - 20 different of such probe pairs (Chaudhuri, 2005). The intensities values of these pairs caused by the hybridization of labeled sample cDNA molecules are combined to get an overall expression level for each gene (see chapter 1.4.1). In addition to the probe sets, which represent genes of the organism of interest, oligonucleotide arrays contain additional probe sets that correspond to genes of other organisms; these spots should help to measure unspecific hybridizations and can also be used for the normalization of different arrays. In contrast to the two-color cDNA arrays, to an oligonucleotide array only the cDNA population of one labeled sample is hybridized at a time.

In summary due to the increased sequence length of cDNAs, a slight advantage of the cDNA arrays could be the greater specifity of the individual spotted probes compared to the shorter probe sequences of the oligonucleotide arrays (Duggan *et al.*, 1999). On the other hand, the lower specifity of single probes on oligonucleotide arrays is compensated by the circumstance that each gene is represented by several oligonucleotide probes covering different parts of the same RNA molecule (see above). In addition, since the used oligonucleotides all have the same number of nucleotides, length-dependent effects that can occur for the probes of cDNA arrays are avoided. Furthermore, cDNA arrays bear another major disadvantage: the greater variability in spot quality and composition as compared to the uniform spot composition of oligonucleotide arrays such as the GeneChip® platform (Knudsen, 2002). In addition, the above mentioned increased sequence length of the spotted cDNAs can lead to multiple contacts of the probe DNA to the slide's surface and to intra-molecular hybridizations causing double-stranded fragments. Therefore, cDNA probes are not as accessible to the sample DNA as oligonucleotides (Duggan *et al.*, 1999). One advantage of the considerably lower spot variability of oligonucleotide arrays is that, after preprocessing the raw data, measurements from different chips are directly comparable to each other. In contrast, for cDNA arrays this is not the case and, therefore, from the two differently labeled samples hybridized to the same cDNA chip (see above) usually one sample is a common control condition which has to be present on each array to enable the comparability of different cDNA chips. Altogether, for organisms with fully sequenced genomes the above mentioned advantages and the fact that

more differentially expressed genes were identified in studies using oligonucleotide than with the cDNA-based platforms (Yauk *et al.*, 2004; Naidoo *et al.*, 2005, see), argue for the use of oligonucleotide microarrays where possible, available and affordable. However, for organisms for which only a minimal genome sequence is available and which are, therefore, not fully sequenced the fabrication of cDNA arrays can be the only alternative (Hardiman, 2004).

1.2.3 Designing an Microarray Experiment, Experimental Workflow and Fields of Application

After reviewing the basic principles of the microarray technology and the available platforms, in the following, a broad overview on the workflow of a typical microarray is given (Fig. 2). The focus is on expression data derived from prokaryotes, since such data were analyzed later on in this thesis (chapters 3.2, 3.3 and 3.4). In short, a microarray-based expression profiling experiment, independent of the used platform, can be divided into the following three stages:

1. Planning and preparation of the experiment
2. Execution of the experiment
3. Processing and analysis of the data

Planning and preparing an experiment (1^{st} stage) comprises the formulation of a biological question which should be posed to the system and its realization leading to the design of the actual experiment (first and second box of Fig. 2). The latter step is crucial since all subsequent steps depend on the initially chosen experimental design. For both, cDNA and oligonucleotide arrays, there are two different experimental design classes that are mainly ascribed to the two following underlying processes that determine mRNA abundance in prokaryotes. Firstly, the rapid alterations of relative mRNA levels due to varying regulatory signals and secondly the dynamic control of relative mRNA abundance by the rates of transcription and mRNA turnover (Conway and Schoolnik, 2003). Hence, the two resulting design classes for the detection of differences in mRNA abundance are, firstly, either to measure the response to multiple stepwise alterations of the conditions or, secondly, to determine the values for two conditions. The first class represents more complex experiments such as time courses, which are only shortly addressed (chapter 3.2.3). The second class represents the widespread classical comparison of two conditions, which occurs in many places throughout this work. Therefore, this design class is explained in more detail below. The comparison of two conditions can have three distinct variants:

- Alterations in the growth parameters, e.g. aerobic versus anaerobic growth (Ye *et al.*, 2000) or growth in minimal versus rich medium (Tao *et al.*, 1999)
- Treated versus untreated cultures: Exposure to substances that drastically change the growth behavior or induce global regulatory networks, e.g. the addition of acetate (Arnold *et al.*, 2001) or DL-norvaline (Eymann *et al.*, 2002) to the growth medium
- Wildtype versus mutant strains, e.g. the knockout of a global transcriptional regulator (Hung *et al.*, 2002; Salmon *et al.*, 2003, 2005)

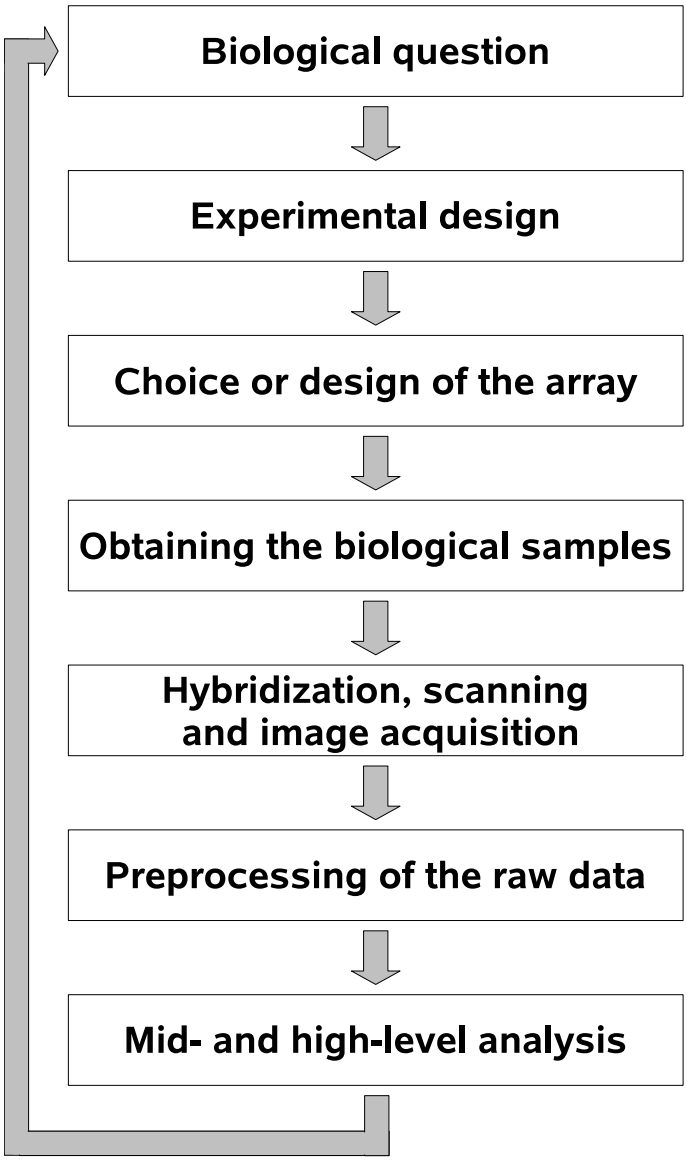


Figure 2: Overview on the workflow of a DNA microarray experiment used in high-throughput expression profiling. The starting point of each experiment is the formulation of one or several biological questions (1^{st} step). Then, the experimental design follows including the choice of cell types or strains as well as the experimental or cultivation conditions and the number of replicates (2^{nd} step). Afterwards, an appropriate microarray platform is chosen, e.g. a genome-wide oligonucleotide array, or synthesized (3^{rd} step). In the next step (4^{th} step) the biological samples are prepared and labeled, e.g. cDNAs obtained from total RNA of the bacterial cultures of interest. The sample is then hybridized to the microarray, the emitted fluorescence detected by a laser and the raw image data are stored with the help of a computer (5^{th} step). In order to get the gene expression levels from the raw image measurements, the data have to be preprocessed (6^{th} step). Then, the mid- and high-level analyses can be performed which include the identification of differentially expressed genes, clustering of the genes or experimental conditions and the detection of significantly affected biological processes (7^{th} step). The figure was adapted from Causton *et al.* (2003) and Chaudhuri (2005).

Accurately designed microarray experiments can represent valuable tools for elucidating a prokaryote's growth physiology (alterations of growth parameters) or regulatory networks (wild type versus mutant). However, several things have to be taken into account to achieve such an optimal design. For increased interpretability, preferably, only one parameter should be varied per experiment. For example, if the effects of aerobic versus anaerobic growth should be compared, the growth phase should be the same – e.g. both in in mid-logarithmic phase (Conway and Schoolnik, 2003). Furthermore, because of the noise inherent to microarray measurements an appropriate number of replicates – at least 3 (Lee *et al.*, 2000) better 4 or more (Hung *et al.*, 2002) – has to be performed as well as the inclusion of appropriate technical and, particularly, biological replicates (Dharmadi and Gonzalez, 2004). Further information on these and additional topics on experimental design can be found in Ye *et al.* (2001), Churchill (2002), Morrison and Ellis (2003) and Stoughton (2005).

After the comprehensive description of the first phase of the preparation and design of a microarray experiment that finishes with the choice of an appropriate array platform (Fig. 2), the remaining two phases are the execution of the experiment and the processing and analysis of the raw data. They are briefly summarized in the following. The second phase of the actual performance of the experiment includes all necessary laboratory tasks starting with the generation of the biological samples (4^{th} box in Fig. 2). This step comprises, if necessary, the generation of mutant strains, the cultivation of the cells under defined parameters, the isolation of the RNA and the generation of labeled cDNA fragments. The latter are utilized for the subsequent hybridization with the DNA microarrays (5^{th} box in Fig. 2). The experimental phase ends with the scanner-based quantification of the amounts of labeled cDNA hybridized to the probes of the microarray. This step yields the raw image data and, after applying appropriate image recognition and appraisal software, the raw expression data. The last phase, processing and analyzing the data, comprises the obligatory preprocessing of the raw data, which is described in detail in chapter 1.4.1, and the mid- and high-level analysis of the preprocessed data, which are outlined in the chapters 1.4.2 and 1.4.3.

1.2.4 Storage and Bioinformatical Representation of Microarray Gene Expression Data

Finally, the bioinformatical storage and sharing of microarray gene expression data is briefly reviewed. Both, raw and preprocessed microarray gene expression, can be represented as a so-called gene expression matrix. This two-dimensional matrix could be considered as a table, whose rows correspond to the measured genes and whose columns to the individual experimental conditions (performed hybridizations). In the first instance, replicate measurements are not combined (Causton *et al.*, 2003). Several variants of the expression matrix exist with respect to the type of expression data, which can comprise absolute expression levels (in arbitrary units), relative expression levels in relation to a common control condition (ratios) or relative expression levels transformed to logarithms (log ratios). In addition, the matrix can on the one hand contain only data from one experimental series that include related hybridization from the same laboratory, using identical protocols, organism and reference sample and the same array platform. On the other hand it can contain results combined from different experiments. In practice, a mixture of both is common. Then, different sections exist that represent measurements from the same experimental series with a common reference sample (Causton *et al.*, 2003).

One well-known example is the classical yeast data set from Eisen *et al.* (1998) which combines about 80 different experimental conditions. Expression matrices can either be stored using spreadsheet programs as well as in special self-hosted databases, both for internal use, or they can be deposited in public databases e.g. in conjunction with a related publication. Well established examples of such database are the Stanford MicroArray Database (Ball *et al.*, 2005), ArrayExpress from the EBI (Brazma *et al.*, 2006; Parkinson *et al.*, 2007) and Gene Expression Omnibus (GEO) from the NCBI (Barrett *et al.*, 2005, 2007). These are, amongst the Pubmed literature database, valuable sources for expression data. All three databases support and encourage the deposition of expression data compliant with the MIAME standard. MIAME (Minimum Information About a Microarray Experiment) describes the minimal information that is necessary to enable the interpretation of the results of an experiment unambiguously and that are needed to reproduce the experiment (Brazma *et al.*, 2001). Thus, not only the expression matrices themselves containing both, the raw and the preprocessed data, respectively, are stored but also additional information like sample annotation including the experimental factors, the experimental design, the annotation of the microarray (e.g. gene identifies, positions on the array) and the laboratory as well as data preprocessing protocols (e.g. method of normalization).

1.3 Bioinformatical Representation of Biological Data

1.3.1 Biological Databases

A huge amount of data was generated by the new high-throughput technologies in bioscience during the past two decades, especially through the sequencing of whole prokaryotic and eukaryotic genomes, which pioneered this development (chapter 1.1). Since conventional methods, e.g. storage as text files, are not well suited to cope with these large amounts of data, database management systems are regularly applied for this purpose. The above mentioned mounds of sequence data are stored in publicly accessible databases such as the EMBL Nucleotide Sequence Database (Kulikova *et al.*, 2007), GenBank (Benson *et al.*, 2007) and DDBJ (Sugawara *et al.*, 2008). These three databases contain nucleic acid sequences that were deposited from scientists all over the world. Meanwhile, they constitute the largest and best known primary databases in this field (Hansen, 2001; Navarro *et al.*, 2003). Primary databases form one group of biological databases which represent repositories of large, redundant high and low quality data sets of DNA or protein sequences – e.g. uncurated genomic sequences – without any prior filtering or additional annotation (Hansen, 2001; Luscombe *et al.*, 2001; Navarro *et al.*, 2003). In contrast, secondary databases, which represent the other broad group of biological databases, are based on filtered and interpreted non-redundant sequence information and additional annotations, for which reason they are also referred to as deduced databases (Hansen, 2001; Kanehisa and Bork, 2003). Therefore, some of these databases were regarded as containing more valuable information and have accumulated not only data but biological knowledge in contrast to most primary databases (Kanehisa and Bork, 2003). But, before going into detail and alleging examples of secondary databases, it has to be emphasized that the above mentioned three primary databases contain all available genome raw sequences, and, therefore, provide the basis for all other biological databases, both primary and secondary databases.

In the meantime a large number of hundreds of secondary databases exists (see also

Table 1: Assortment of well-known curated biological databases.

Database name	**Subject area**	**Internet address** (http://...)
UniProt [1]	protein sequences	www.expasy.uniprot.org
PDB [2]	3D structural data	www.rcsb.org/pdb, www.wwpdb.org
CSD [3]	3D structural data	www.ccdc.cam.ac.uk/products/csd
PRODORIC [4]	prokaryotic genomes, operons, TFBS and TRNs	www.prodoric.de
RegulonDB [5]	*E. coli* genome, operons, TFBS,	regulondb.ccg.unam.mx
TRANSFAC [6]	eukaryotic TFBS	www.gene-regulation.com/pub/databases.html
TRANSPATH [7]	eukaryotic signal transduction	www.gene-regulation.com/pub/databases.html
KEGG LIGAND [8]	enzymes, chemicals and biochemical reactions	www.genome.jp/kegg/ligand.html
KEGG PATHWAY [8]	metabolic pathways	http://www.genome.jp/kegg/pathway.html
BRENDA [9]	enzymes, chemicals, biochemical reactions	www.brenda-enzymes.org
IntAct [10]	protein-protein interactions	www.ebi.ac.uk/intact
STRING [11]	protein-protein associations	string.embl.de

References:

1: Wu *et al.* (2006), 2: Berman *et al.* (2007), 3: Allen and Taylor (2004)
4: Münch *et al.* (2003, 2005), 5: Salgado *et al.* (2006), 6: Matys *et al.* (2003, 2006)
7: Krull *et al.* (2006), 8: Kanehisa *et al.* (2006), 9: Barthelmes *et al.* (2007)
10: Kerrien *et al.* (2007), 11: von Mering *et al.* (2007)

the annual database issues of the journal *Nucleic Acids Research*) with different contextual foci and width of scope as well as varying detailedness and update frequency. As mentioned above, there are some very useful databases which are well and regularly maintained by manual effort and which, therefore, contain a lot of non-redundant high quality data (see also Kanehisa and Bork, 2003). Examples of these databases are given in Table 1 and below. With the help of such databases and bioinformatic tools new biological knowledge can be generated. This knowledge includes for example:

1. the prediction of protein functions using sequence similarities and identified conserved protein domains
2. the reconstruction of whole signal transduction networks from individual signal transduction pathways
3. the deduction of metabolic pathways and networks combining enzyme-catalyzed biochemical reactions
4. the reconstruction of protein-protein interaction networks and molecular complexes from single (binary) protein-protein interactions
5. the reconstruction or prediction of transcriptional regulatory networks (TRN) and transcription factor binding site (TFBS) from genome and proteome sequences

For example, predicted protein functions (point 1) can be obtained utilizing the UniProt knowledgebase (Tab. 1), which offers both, non-redundant sets of curated experimentally verified proteins with high-level annotation (UniProtKB/Swiss-Prot database) and computer-annotated translations of coding sequences from EMBL nucleotide sequence entries (UniProtKB/TrEMBL database containing entries not yet integrated in Swiss-Prot). Signal transduction networks (point 2) can be, amongst others, reconstructed and visualized with the help of the TRANSPATH database about eukaryotic signal transduction events, which provides information about signaling molecules, their reactions

and the pathways these reactions constitute (Krull *et al.*, 2006). Metabolic networks (point 3) can be modeled and visualized by either using KEGG LIGAND and PATHWAY database (Kanehisa *et al.*, 2006) or the comprehensive BRENDA enzyme information system (Barthelmes *et al.*, 2007), which both contain chemical compounds, the corresponding biochemical reactions and enzyme information. For the reconstruction of protein-protein interaction networks and protein-containing molecular complexes (point 4), the IntAct database (Kerrien *et al.*, 2007) might be taken as data source. It stores pair-wise protein-protein interactions. Finally, for the construction and prediction of transcriptional regulatory networks (TRN) two databases that contain a comprehensive collection of transcription factor binding site (TFBS) data and information on the corresponding transcriptional regulators can be used. The TRANSFAC database, which focuses on eukaryotes, e.g. the model organisms human, mouse, rat and thale cress (Matys *et al.*, 2006), as well as the PRODORIC database, which concentrates on prokaryotic organisms (Münch *et al.*, 2003). Furthermore, PRODORIC contains a very extensive compilation of genome sequences from most completely sequenced archaea and bacteria, which comprises also the classical prokaryotic model organisms *B. subtilis*, *E. coli*, and *P. aeruginosa*. Amongst TFBS, the database contains promoters, operons, regulons and ribosome-binding sites as well as some transcriptome data. Besides gene regulatory information, PRODORIC collects also metabolic reactions and signal transduction pathways. Furthermore, it offers a basic non-hierarchical annotation of the genes based on the COG classification (cluster of orthologous genes, see Tatusov *et al.* (2001); its development has, however, ceased in the meantime). Thus, PRODORIC represents an important integrative knowledge base for systems biology approaches applied to prokaryotes. In this context, the PRODORIC database was also expanded (Münch *et al.*, 2005) to include the Gene Ontology functional classification system (Gene Ontology Consortium, 2006), which is outlined in the next chapter (chapter 1.3.2). In addition, during the course of this thesis support was given in the development of another secondary database on systems biology of prokaryotes, the SYSTOMONAS database about molecular networks in pseudomonads (Choi *et al.*, 2007).

1.3.2 Classification Systems and Biomolecular Networks Used in Bioinformatics

Classification Systems:
Classification systems are used to structure biological knowledge and have a long tradition in biology, in particular, in phylogenetic systematics, where they are common since several hundreds of years (see e.g. the "Systema naturae" publication of Linné from the year 1735). Amongst the field of phylogenetics, which tries to classify organisms according to their lineages, in recent times additional biological disciplines have adopted the concept of classification such as, for example, the grouping of proteins or enzymes in particular, according to their sequence similarity, shared domains or functions. Usually, these comprise systematically structured classifications of biological knowledge domains, which are not only human-readable, but can, if designed for this purpose, also be processed by computers, which makes them suitable for a bioinformatics analysis. Examples of such classification systems are:

1. the NCBI taxonomy tree (Wheeler *et al.*, 2000), which is a curated collection of names and classifications of all organisms present in the GenBank database (Benson

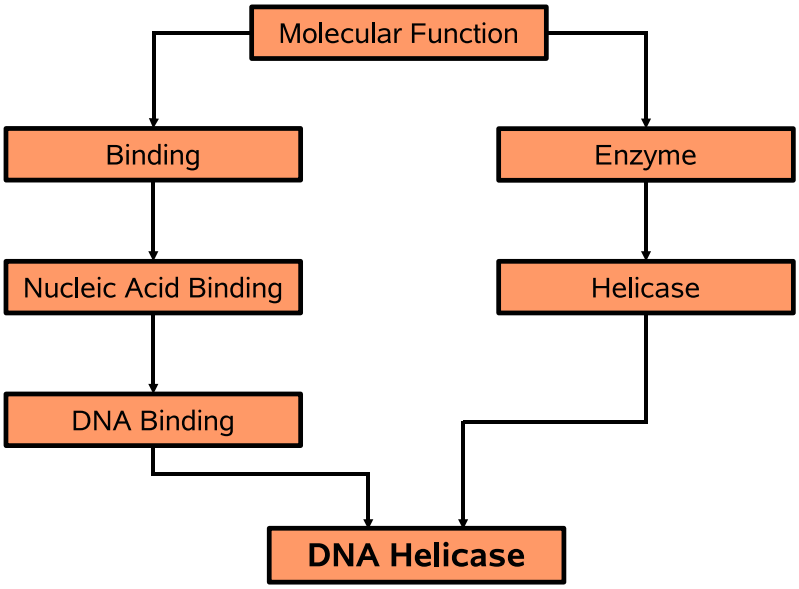


Figure 3: Example of a Gene Ontology term with more than one parent node. A gene product with the *Molecular Function* (MF) '*DNA Helicase*' is both, a DNA binding protein (left path coming from MF) and an enzyme with its specific activity, in this case the separation of two DNA strands (helicase, right path coming from MF).

et al., 2007)

2. the BRENDA tissue ontology (Schomburg *et al.*, 2004), a hierarchical classification for sources and tissues where enzyme are derived from or located in

3. the EC system of the Enzyme Nomenclature Committee (Webb *et al.*, 1992), which groups the enzymes by the reactions they catalyze using four-digit EC numbers

Some but not all of these systems represent so-called ontologies - from the examples above only the BRENDA tissue ontology does. The conception of an ontology originally comes from the field of philosophy, and ontologies have often been used for the description of all entities within a certain knowledge domain and all relationships between them. In the context of computer and information science an ontology defines "a set of representational primitives with which to model a domain of knowledge or discourse" (taken from Gruber, 2008). These representational primitives are normally classes, attributes and relationships among the class members, whereas a predefined vocabulary is used (Gruber, 1993, 2008). Thus, to put it briefly, an ontology used in bioinformatics is composed of two building blocks: 1) a set of well-defined distinct terms and 2) all relationships that exist between these terms. Main advantages of ontologies are their strict vocabulary and the predefined types of relationships between the terms, which allows for a precise human-readable description and the application of axioms for the computer-based analysis.

Gene Ontology Classification System:
An ontology widely used in biosciences is the Gene Ontology (GO), which was also utilized in this work for the high-level analysis of microarray gene expression data (see chapter 1.4.3). It represents a well structured functional classification system of genes and gene products using a strict vocabulary and an unambiguous definition of each vocabulary item

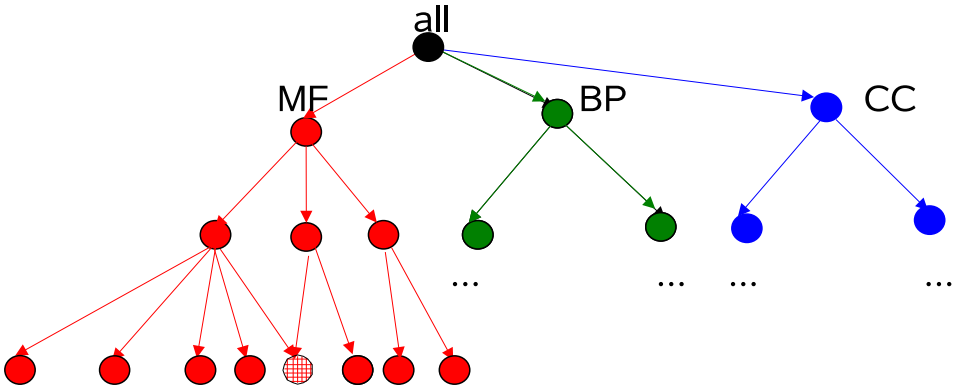


Figure 4: Overview on the structure of the Gene Ontology (GO) which is a rooted directed acyclic graph (DAG). The root node is named '*all*' and represents the predecessor of all GO nodes. On the next level, three nodes representing different aspects of GO are found: *Molecular Function* (MF, red color), *Biological Process* (BP, green color) and *Cellular Component* (CC, blue color) follow. These nodes are themselves the root nodes of three non-overlapping sub-ontologies. Since GO is structured as a DAG, there are also nodes with more than one parent node (see the shaded red node).

(Ashburner *et al.*, 2000; Gene Ontology Consortium, 2006). In addition, it is applicable for every organism, from archaea and bacteria to higher eukaryotes. GO was designed as a rooted directed acyclic graph (DAG) and, therefore, consists of a set of nodes, which represent the biological terms, and a set of direct edges, which represent the hierarchical parent-child relationships between these terms. Thus, child nodes are further specializations of their parent nodes. For clarification a typical example of this issue is shown in Figure 3. Furthermore, the DAG structure implies that a) no path in the graph starts and ends at the same node (acyclic graph) and b) a node can have more than one parent node (Fig. 4). This stands in contrast to tree-based classification systems such as the NCBI taxonomy tree, in which each node has at most one parent node. The area of knowledge (see ontology definition above) of GO, that is the functional description and annotation of genome encoded genes and proteins, was further sub-divided by the Gene Ontology Consortium into the following three domains: 1)*Molecular Function* (MF), 2) *Biological Process* (BP) and 3) *Cellular Component* (CC, see Ashburner *et al.*, 2000). They also constitute direct child nodes of the root node and, thus, represent the second level of GO. In addition, they are themselves root nodes of three sub-ontologies, which are non-overlapping since they share no edge. A detailed explanation of the three sub-ontologies is given in the following (adapted from `http://geneontology.org/GO.doc.shtml`):

- *Molecular Function* refers to the activity or activities that a gene product can perform at the molecular or biochemical level. Only the activity itself is specified, but not the time point of the action, its context or whether it is performed by the gene product alone or by complexes of gene products. Examples of more general terms are *binding activity*, *catalytic activity*, *transcriptional activity* and *transporter activity*.

- A *Biological Process* specifies the broader context to which gene products can add to. Normally, this is achieved by events accomplished by one or several MFs that have taken place consecutively. In order to distinguish it from an MF, the number

of distinct steps is critical: a BP reflects more than a single event. Examples of more general terms are *regulation of transcription (DNA-dependent)*, *signal transduction* and *regulation of cell motility*. More specific terms are, for example, *purine metabolism*, *anaerobic respiration* and *glyoxylate cycle*. It has to be stated that a BP is not the same as a pathway since dynamics or dependencies that would be necessary for the complete description of a pathway are not provided by GO, up to now.

- As the name suggests, a *Cellular Component* represents the location within a cell or the extracellular region in which a gene product is active or can be found. This includes on the one hand sub- and extracellular locations such as *nucleus*, *bacterial nucleoid*, *Golgi apparatus* and *outer membrane* (of Gram-negative bacteria). On the other hand, molecular complexes and groups of gene products such as *ribosome* and *proteasome* are specified, too.

Amongst the three types of sub-ontologies, a further feature of GO is that nodes are connected by edges which represent one of two types of relationships: *is_a* and *part_of* (Harris *et al.*, 2004). While *is_a* describes a straight-forward class-subclass relationship, the *part_of* relationship is a bit more complex: for example, when A *part_of* B is valid, A is always, when present, part of B; however, A needs not always to exist.

GO is continuously growing and updated, which leads to a) the addition of new nodes (terms), b) the deletion of old obsolete nodes, c) the merging of nodes and d) the renaming of nodes. During this process, it is made sure by the Gene Ontology Consortium that the changes of existing nodes are as small as possible and renamed terms keep their accession numbers. As a result of the continuous and well-curated update process, the number of nodes and edges of GO is steadily growing. A representative inventory of these quantities is shown in Table 2. Another effect is the broad acceptance as a standard in the field of gene annotation by the scientific community. This is also reflected by the increasing number of publications citing GO.

Finally, one aspect should be mentioned which makes GO especially valuable as a resource for knowledge-based analysis of gene expression data (chapter 1.4.3) since it allows the assignment of gene products to the GO terms. This is also called annotation of gene products and is in principle independent from the structure of GO. Thus, like a gene product can have multiple functions or can take place in more one than biochemical pathway, one gene can be assigned to several GO nodes from all three sub-ontologies. A comprehensive collection of assignments is provided by the Gene Ontology Annotation

Table 2: Gene Ontology in numbers (status: end of November 2005). The overall number of GO nodes and edges between them is given as well as the number of the GO nodes that belong to each of the three sub-ontologies.

GO nodes representing a *molecular function*	7866
GO nodes representing a *biological process*	10440
GO nodes representing a *cellular component*	1775
remaining GO nodes such as root node and relationship types	5
GO nodes (total)	20080
egdes (total)	29200

(GOA) project (Camon *et al.*, 2003, 2004). It offers high-quality electronic and manual annotations (Camon *et al.*, 2004) to proteins from the UniProt knowledge base (Wu *et al.*, 2006). Every gene product to GO node assignment is provided with an evidence code and a code that indicates the type of annotation – e.g. electronic or manual annotation – as well as the source, e.g. a literature reference, an external database or a computational method. GOA, like all other GO annotations, assigns a gene product only to the most specific GO node available. But, due to the true path rule the assignment can be propagated to all parent nodes, since they represent generalizations of the more specific GO nodes with the direct assignment (see Gene Ontology Consortium, 2001). Altogether, GOA contains annotations for about 60,000 species and is, therefore, the most comprehensive provider of annotations to GO and regarded as the *de facto* standard in this field.

Biomolecular Networks:
While classification systems normally represent deliberately structured hierarchies for ordering items from different levels of abstraction and can be modeled as tree or acyclic graphs, a key feature of biomolecular networks is the occurrence of recurring circuit elements (Alon, 2003; Ideker *et al.*, 2002). Thus, in contrast to classification systems, biomolecular networks must almost always be modeled as cyclic graphs. Other typical features of biological networks are their modularity and their robustness to component tolerances (see Alon, 2003, for details). The four basic types of biological networks, a) signal transduction networks, b) metabolic networks, c) protein-protein interaction networks and d) transcriptional regulatory networks, have been exemplified above in the context of biological databases (chapter 1.3.1). Generally, these networks are abstract representations of biological systems or parts of them, and grasp most of their characteristic properties (Alon, 2003). The computational representation and modeling of such networks depends on their type, the amount of information available and the intended degree of detailedness. Molecules are represented by the nodes of the graph and their interactions, such as protein-protein or protein-DNA interactions, by the edges (Alon, 2003). These edges can be directed or undirected and, in addition, be unweighted or weighted. Weighted edges represent different strengths of interactions, which are specified by the number assigned to them.

In the case of gene regulatory networks (TRNs) the nodes usually stand for genome-encoded proteins, which also includes the subset of the transcriptional regulator proteins. Directed edges point from the transcriptional regulators to the gene products encoded by the target genes and therefore, the direction of edges corresponds to the information flow from the transcription factor to the gene that it regulates (Barabasi and Oltvai, 2004). Since prokaryotic genomes are organized as operons, which contain one to several co-transcribed genes under the transcriptional control of a common promoter (Madigan and Martinko, 2006), all genes belonging to the same operon have to be included as target genes (see also Shen-Orr *et al.*, 2002). Positive and negative transcriptional regulation can be modeled by an edge weight or sign, but can also be omitted for the sake of simplicity. Altogether, a certain transcription factor node, all its (known) target gene nodes and the connecting edges represent a regulon, which is defined as group of genes regulated by the same transcriptional regulator or stimulus. These regulons are the building blocks of the modeled regulatory network and can be used for the computational analysis such as in the knowledge-based analysis of gene expression data, which is described in chapter 3.4.

1.4 Preprocessing and Knowledge-based Analysis of High-throughput Gene Expression Data

The analysis of microarray expression data is a multi-level task, which can be basically divided into the three parts: the low-, mid- and high-level analysis (Fig. 2). The low-level analysis, also known as preprocessing, is the obligatory first step and is necessary to transform the raw data derived from a microarray measurement into relative gene expression levels (see also Zareparsi *et al.*, 2004). Based on the latter and taking appropriate replicate measurements into account, the statistically more robust probabilities of differential expression are computed during the mid-level analysis, whose conduction is recommended. Low- and mid-level methods of analysis are reviewed in the next two chapters (chapter 1.4.1 and 1.4.2). The high-level analysis, finally, comprises more in-depth data mining methods such as clustering of the expression data or the interpretation in the context of biological classification systems and other sources of biological knowledge (Zareparsi *et al.*, 2004). The latter is the focus of chapter 1.4.3.

1.4.1 Low-level Analysis of Microarray Expression Data

The goal of the low-level analysis, which is also called preprocessing, is to adjust for any non-biological variability or experimental noise inherent to the sample preparation and the microarray technology itself. It can be subdivided into the following 6 subsequent steps (see Gentleman *et al.*, 2005): 1) image analysis, 2) data import, 3) background correction, 4) normalization, 5) summarization and 6) quality assessment. During image analysis, probe intensity data are computed from the scanned and digitized microarray images after the identification of the spots and the quantification of their pixels. During the data import gene names, gene identifiers, the array layout, sample annotations and other data relevant for the subsequent steps are collected – sometimes from several different files or databases (Gentleman *et al.*, 2005). For both tasks several software tools exist, which are normally provided by the manufacturer of the microarray platform. They perform these steps largely in an automatic manner. Therefore, they are not further outlined here. More in-depth information can be for example found in Yang *et al.* (2002) for cDNA array platforms and Schadt *et al.* (2001) for oligonucleotide array platforms.

The third step, the background correction, also known as background subtraction, is necessary to account for non-specific hybridizations and for the noise of the optical intensity detector. Thus, mainly accurate measurements of the hybridizations are obtained (Gentleman *et al.*, 2005). The normalization step makes different microarray hybridizations comparable to each other by adjusting for miscellaneous types of inter-chip variation due to non-biological causes such as different labeling efficiencies or unequal quantities of initial RNA, differences in the sample preparation or in other laboratory conditions etc. (Gentleman *et al.*, 2005). During the removal of these systematic sources of variability, the biological variation is retained (see also Oberg *et al.*, 2006). Finally, summarization is required for microarray platforms with probe redundancy, in which transcripts are represented by more than one spot like the oligonucleotide array platform GeneChip® from Affymetrix (see chapter 1.2.2), where a transcript is represented by several oligos. In the end each transcript obtains a single summarized expression value which is proportional to the quantity of the corresponding RNA (Gentleman *et al.*, 2005). Preprocessing in the narrow sense of the word comprises the three mentioned main steps background correction, normalization and summarization (see also Irizarry *et al.*, 2006). Therefore, they

are referred to in more detail below in the context of different preprocessing algorithms. At the same time the focus is on the GeneChip® microarray, since raw data from this platform had to be preprocessed in the thesis at hand (chapter 3.3.1).

The last task in a low-level analysis is the quality control. It serves for the detection of diverging measurements that are beyond the acceptable level of random fluctuations (taken from Gentleman *et al.*, 2005). This includes the comparison of replicate hybridizations with diagnostic plots such as scatter and MA plots or the computation of correlation coefficients.

Before proceeding with the introduction of common preprocessing methods and algorithms, the statistical foundations of them, including a simple error model, are briefly presented as well as the two principal types of preprocessing approaches.

Statistical foundations and error model:
The above described effects of variations in microarray measurements can be divided into two types: systematic effects and a stochastic component, that represents the noise (see Gentleman *et al.*, 2005). While the systematic effects apply simultaneously to the majority of hybridized spots, e.g. all probes of a chip, the stochastic components occur randomly and lack regularity. Therefore, stochastic models are often employed for preprocessing, since they are well suited for determining the noise component, but also allow to compute systematic effects in good approximation. The following equation (Eqn. 1) describes a general-purpose error model for the intensity value I of a single hybridized probe on a particular array slide (taken from Gentleman *et al.*, 2005):

$$I = B + \alpha S \tag{1}$$

Here B represents the background noise, which results e.g. from non-specific hybridizations. Therefore, its value is random. In contrast, S corresponds to the signal caused by the specific binding of the probe, which is amplified by the factor α. S is also a random variable and consists of three different components: the true intensity value that is proportional to the quantity of hybridized sample, the measurement error and probe-specific effects (Gentleman *et al.*, 2005). This circumstance is described by the following equation (Eqn. 2), which allows the computation of S at the logarithmic scale:

$$log(S) = \theta + \phi + \varepsilon \tag{2}$$

θ is the abundance, ϕ corresponds to the probe effect and ε represents the error term. Equation 2 is the so-called additive-multiplicative error model, which was introduced by Rocke and Durbin (2001) as well as Ideker *et al.* (2000).

Types of preprocessing approaches:
Whether taking the additive-multiplicative error model into account or not, two principal approaches of preprocessing microarray expression data exist according to the order of the individual tasks: stepwise and integrated approaches. While stepwise approaches perform the three preprocessing steps background correction, normalization and summarization successively, integrated approaches perform the analysis at once by combining the different tasks (Gentleman *et al.*, 2005). Stepwise approaches bear the advantages that the gene expression matrix is built up in a modular manner, that different types of background correction and normalization can be more easily joined and that they are more easily to compute than integrated ones (Wu and Irizarry, 2007). On the other hand, one drawback of the stepwise procedures is that every task is optimized independently

from the others, which can cause sub-optimal overall results (Gentleman *et al.*, 2005). This is not the case for integrated approaches. One example of an integrated approach is the vsn method (Huber *et al.*, 2002) which combines the background correction and normalization step (see below).

The outcome of each preprocessing procedure, irrespective of the type of approach, is a gene expression matrix (chapter 1.2.4), which can be used for subsequent analyses.

Overview on common preprocessing methods:
As mentioned above, the development of preprocessing methods for the computation of gene expression levels has become a field of active research (Irizarry *et al.*, 2006). For this purpose, a variety of different preprocessing methods is now available for the microarray technology in general and for the Affymetrix GeneChip® platform in particular, which is widely applied in microbiological research (Cope *et al.*, 2004). On the one hand this large choice offers the advantage to select a specific method suited for a particular question and data set. On the other hand, it is not always obvious or easy to identify the best method in a special context (see Zhang *et al.*, 2005). Therefore, several benchmarks have been conducted that evaluated the performance of different preprocessing methods using the same raw data sets (Cope *et al.*, 2004; Irizarry *et al.*, 2006; Shedden *et al.*, 2005; Millenaar *et al.*, 2006; Seo and Hoffman, 2006). The essential precondition was that for the selected benchmark data sets, which comprised dilution and spike-in experiments, the expected outcome is known in advance.

In the following, the most common preprocessing methods are described and, afterwards, the strengths and drawbacks of them are summarized based on the results of the benchmarks. The methods comprise MAS5 (see Affymetrix Inc., 2001, 2002), dChip (Li and Wong, 2001a,b), rma (Irizarry *et al.*, 2003a,b) and vsn (Huber *et al.*, 2002, 2003). These nicknames were used throughout this work and specify particular combinations of a background correction, normalization and summarization method.

MAS5:
The MAS5 method was developed by Affymetrix, the vendor of the GeneChip® platform. For the background correction, MAS5 uses a zone-based algorithm and offsets the perfect matches probes (PM) against their corresponding mismatch probes (MM, see chapter 1.2.2 and Affymetrix Inc. (2002)). During this course, the microarray is divided into areas (by default 16), to take spatial background drifting into account (see Affymetrix Inc., 2002; Zhang *et al.*, 2005). The cells of each such a zone are sorted by their average intensities and those with the lowest values, normally the 2^{nd} percentile (the lowest 2%), are regarded as background of this area. For the computation of the actual background value of a point on the array, a weighting function is used that considers the distance between it and the centroid of the respective zone. The occurrence of negative values after subtraction of the local background is prevented by using a small preset threshold value. Afterwards, the frequency distribution of the values of the PM and the MM probes have a shape similar to a normal distribution (see also Zhang *et al.*, 2005). Subsequently, the PM probes are adjusted using the MM probes by the ideal mismatch algorithm to account for non-specific binding. The simple subtraction of MM intensity values from the PM intensity values, as used in earlier version of MAS algorithms, generated many negative values due to the fact that around 30 % of the MM probes have higher intensities than the corresponding PM probes (Gentleman *et al.*, 2005). To circumvent this, the ideal mismatch algorithm allows the subtraction only when $MM > PM$ is true and otherwise

the value of PM is set to that of the MM probe and a small constant is added to the PM intensity, afterwards. Then, the adjusted intensities of the PM probes are transformed to logarithms to stabilize the variances (see Affymetrix Inc., 2002). The normalization of the MAS5 method consist only of a linear scaling step using a trimmed mean. It is the simplest form of normalization and assumes a Gaussian distribution for the intensity values of an array. It just shifts this distribution to a new center by multiplying the signal values with a constant scaling factor. This approach can become problematic, when chip to chip differences in the intensity value distributions are large (Zhang *et al.*, 2005). In contrast to many other algorithms, scaling in MAS5 is performed with already summarized gene expression levels. For the summarization, the one-step Tukey's biweight method is used. Here, a so-called biweight estimator is taken for the computation of a robust mean value analogous to the arithmetic mean or the median, whereas the signal intensity is the anti-log of the computed value (see Zhang *et al.*, 2005). In general, it should be stated that the MAS 5.0 method performs separate single chip preprocessing steps and does not take information across the chips of a multi-array experiment into account.

dChip:
The dChip method computes a model-based expression index (MBEI) for each probe set and considers information across the individual arrays analyzed. The method can either take the intensities of the MM probes into account or leave them out. In the following, the second version is described that makes solely use of the PM probe intensities. A background subtraction step in the narrow sense of the word is not performed (see also Zhang *et al.*, 2005). In the normalization step of a multi-chip experiment, an array with medial overall intensity is chosen and serves as the baseline chip. The other arrays are to be normalized at the level of probe intensities against this reference chip (Saviozzi and Calogero, 2003). The aim is then to base the normalization on probe intensities that represent genes which are not differentially expressed (Li and Wong, 2001b). This set of genes is also called the invariant set. They are identified by an algorithm that assumes similar intensity ranks of these genes on different arrays. For this reason, the intensities of all PM probes are ranked and afterwards compared with the baseline chip to determine those with similar ranks (see above). Thus, a new and presumably different invariant set is computed each time when the composition of arrays in the experiment changes. The summarization algorithm of the dChip method considers the empirically determined fact that the variation of a particular probe can be much smaller across different chips than the variance across probes within a probe set of the same array. This argues for a considerable probe affinity effect (Saviozzi and Calogero, 2003). The expression index y_{ij} of array i and probe j can be computed according to a multiplicative model, which is expressed by the following equation:

$$y_{ij} = PM_{ij} - MM_{ij} = \theta_i \phi_j + \varepsilon_{ij} \tag{3}$$

Here (Eqn. 3), θ_i denotes the model-based expression index (MBEI) of the array and ϕ_j the probe-sensitivity index for a specific probe, whereas ε_{ij} corresponds to the random error. This model is analogous to the generic error model shown above (Eqn. 2). The model fitting procedure (least square fit) iteratively generates more reliable estimates for the true expression measure, the MBEI θ. Although the first model was based on the difference between PM and MM, a more recent version of the algorithm was developed that only uses PM intensities (MM term of Equation 3 is omitted). This is also known as

the PM only version (Li and Wong, 2001b). The exclusion of MM probes even increases the performance of the method and offers better estimations of the expression levels (Li and Wong, 2001b). In comparison to the MAS5 method, the dChip method can better cope with weakly expressed transcripts, since it lowers the variation of the expression intensity estimate.

rma:
As the new version of the dChip algorithm and other MBEI methods, the rma method also does not take MM probes into account since they do not reliably represent the background signals. Both dChip and rma use statistical models instead. The rma (robust multi-array average) method fits a robust linear model to the probe-level data. It preprocesses each array of the experiment in context with the others. The model that is exploited for the background correction is based on the assumption that the PM intensity distribution can be divided into an exponentially distributed signal component S and a normally distributed noise component N (see Zhang *et al.*, 2005). The three parameters of this distribution, the mean α of the exponential distribution (S), the mean μ and the standard deviation σ of the Gaussian distribution (N) are estimated using the convolution product of S and N and a density kernel estimation (see Zhang *et al.*, 2005). The normalization used by the rma method is the so-called quantile normalization. It adapts the distributions of probe intensities between the different chips to be same for each array. The underlying concept is the two-dimensional quantile-quantile plot, which indicates that two distributions are the same when a diagonal line is present. This approach was expanded to the n-dimensional space, whereas n corresponds to the number of arrays in the experiment (see Saviozzi and Calogero, 2003). For this purpose, first the highest PM intensity values (transformed to the logarithm) of each array are identified and averaged. Then, these averaged values are used to replace the individual ones. This step is repeated for the next largest intensity value and so on. While as a result of the normalization a probe could have the same value across all chips, this is normally not the case for a particular gene since averaging occurs at the probe level and a gene is represented by many PM probes (Zhang *et al.*, 2005). Finally, for the summarization a median polish method is employed. It is based on the observation that each probe intensity value is the sum of the true gene expression level, the corresponding probe affinity across all arrays analyzed, and a random error term. This can be described by the following equation (Eqn. 4):

$$Y_{ijn} = \mu_{jn} + \alpha_{jn} + \varepsilon_{ijn} \tag{4}$$

$$i = 1, ..., I(chips); j = 1, ..., J(probes); n = 1, ..., N(probesets)$$

Here, Y_{ijn} is the summarized expression statistics, μ_{jn} represents the expression level (log scale, chip-independent), α_{jn} the probe affinity effect and ε_{ijn} a random error term.

vsn:
The method of the variance stabilizing transformation (vsn) integrates the steps of background subtraction and normalization (Huber *et al.*, 2002, 2003). By doing so, information from different arrays can be shared when estimating the parameters for the background correction (see Gentleman *et al.*, 2005). The vsn method is based on the empirical observation that with increasing mean values replicate microarray measurements normally bear a larger variance of their probe intensities. The algorithm uses an affine transformation of the probe intensities and a subsequent inverse hyperbolic sine

transformation, which stabilizes the variance in the whole range of intensities (see Zhang *et al.*, 2005). This transformation is described by the following equation (Eqn. 5):

$$h_i(y_{ki}) = arsinh(a_i + b_i y_{ki});\ arsinh(x) = ln(x + \sqrt{x^2 + 1}) \quad (5)$$

$$i = 1, ..., d(chips); k = 1, ..., n(probes)$$

In Equation 5, $h_i()$ is the variance stabilizing transformation of the measured raw probe intensity value y_{ki} of probe k on the ith array (Zhang *et al.*, 2005). The parameter a_i and b_i are array-specific and are estimated from the measured intensity values with a robust variant of a maximum likelihood estimation (Huber *et al.*, 2002). The inverse hyperbolic sine transformation is for large (positive) intensity values (x) equivalent to a logarithmic transformation ($\lim_{x\to\infty}(ln(x + \sqrt{x^2 + 1})) \approx log(2x)$). However, in contrast to the logarithm, it is also defined for small intensity values of zero and below (Huber *et al.*, 2002). Therefore, the vsn method can handle both, positive and negative intensity values. As for the rma and dChip method (recent version), the vsn method also solely uses the PM probe intensity values for the preprocessing. Like rma, it uses a median polish approach for the summarization.

Performance and assessment of the presented preprocessing methods:
The chosen preprocessing method normally has impact on the computed expression levels and consequently different genes will be marked as differentially expressed in subsequent analyses (see Millenaar *et al.*, 2006). The overlap of the different algorithms with respect to the outcome ranges from high to lower values – e.g. below 40% (see Millenaar *et al.*, 2006) – , although for the most differentially expressed genes it is often higher than the average overlap. Some features of the preprocessing methods (see the descriptions above) are useful for assessing their performance. For example, one main drawback of the MAS5 algorithm is its usage of MM probes for estimating the background signal, since these show, as mentioned above, in at least one third of the probe pairs have smaller signals than the corresponding PM probes and no direct benefit of the MM probes has been reported so far (Cope *et al.*, 2004). Probably, this partially explains the better performance and lower variance of the PM only algorithms (see also Seo and Hoffman, 2006) such as dChip, vsn and rma. In addition, the normalization of the MAS5 algorithm is improvable compared to the other three methods, since it, unlike the others, does not take information from other arrays into account. Consequently, benchmarks indicated that dChip and rma perform better than MAS5 (see Irizarry *et al.*, 2003a; Cope *et al.*, 2004; Zhang *et al.*, 2005). May be due to the lack of a sophisticated background correction, the dChip method was outperformed by the rma method (Irizarry *et al.*, 2003a). Both, rma and vsn, obtained similar good results in another benchmark (Irizarry *et al.*, 2006). In general, rma is more often used and more frequently cited in the context of microarray data preprocessing than vsn. In addition rma was also recommended, since it performed better than competitive methods in several benchmarks (see Allison *et al.*, 2006). In this work the three PM only methods were applied to bacterial microarray data in a comparative study and only rma was used for the computation of expression levels in another study (chapter 3.3.1).

1.4.2 Mid-level Analysis of Microarray Expression Data

Based on the preprocessed expression levels computed in the low-level analysis, one goal of the mid-level analysis is to identify differentially expressed genes. In contrast to the

usage of an arbitrary fold change threshold value (e.g. two-fold) to identify the interesting genes, the mid-level analysis computes a probability of differential expression (pde) for each gene considering also the variance of the expression values (see Cui and Churchill, 2003; Zareparsi *et al.*, 2004). In the case of a pairwise comparison of two samples (see chapter 1.2.3) conventional t-tests and derivates of them can be used, especially when the error variance of each sample is normal-like distributed or the samples are small. Non-parametric tests such as rank-order test can be applied, alternatively (see Zareparsi *et al.*, 2004). Here, a sample distribution denotes the preprocessed expression levels of all replicate measurements under a given condition. Since the number of replicates and, thus, the sample sizes are often small due to the high costs for a microarray experiment, the standard t-test (see chapter 1.4.3) has a low power. For these reasons, modifications of the t-test are commonly used that, firstly, compute more stable estimates of the variances (see Cui and Churchill, 2003). In order to achieve this, they normally combine data from other genes and, thus, 'borrow' information across genes (see Allison *et al.*, 2006). One example is the significance analysis of microarrays (SAM). It is a modification that adds a small positive constant to the denominator of the gene-specific t-test while pooling all genes for estimating the error variance (Tusher *et al.*, 2001). In a permutation-based approach the significant genes are identified. Another example of a modified t-test represents the regularized t-test, whose implementation is known as CyberT (Baldi and Long, 2001). It makes use of the observation that there is a reciprocal relationship between the variance and gene expression levels and genes with similar expression levels show a similar variance, too (Hatfield *et al.*, 2003). This prior knowledge is used in a Bayesian approach that computes a weighted average of the variance of the gene itself and the pooled variance of genes with similar expression levels. This represents a more robust estimation of the variance for any gene of interest (Hatfield *et al.*, 2003). Both, SAM and the regularized t-test are considered as working equally well (see Allison *et al.*, 2006).

Finally, it should be mentioned that a multiple testing problem arises, since for each gene a statistical test is performed. Therefore, filtering genes according to a predefined significance level requires an adjustment of the obtained p-values. Appropriate corrections of the multiple testing effects such as the Bonferroni (Bonferroni, 1936, see also Bland and Altman, 1995) or the FDR method (Benjamini and Yekutieli, 2001) as well as permutation-based procedures are often integrated in the software packages that offer implementations of the above mentioned modified t-tests.

1.4.3 High-level Analysis of Microarray Expression Data

Overview on the principal approaches

The high-level analysis of microarray gene expression data comprises computational methods that use the gene expression levels – often in form of a gene expression matrix (see chapter 1.2.4). It is obtained from a precedent low- (chapter 1.4.1) or mid-level analysis (see chapter 1.4.2). The methods can broadly be divided into two classes:

- data-driven approaches using the expression data as sole information source
- methods using *a priori* biological knowledge

The first class, data-driven approaches, represents unsupervised methods and is also called class discovery (see Allison *et al.*, 2006). It comprises the classical data mining

techniques such as hierarchical clustering, K-means clustering and self-organizing maps (see Zareparsi *et al.*, 2004). These types of methods were, for example, successfully used for clustering the genes according to their expression levels (Eisen *et al.*, 1998). This allows to identify groups of genes with a similar expression profile throughout the different tested experimental conditions. Other applications include the reciprocal approach of clustering the experimental conditions or clustering both, genes and conditions, at once, which is also called biclustering (see Cheng and Church, 2000; Madeira and Oliveira, 2004; Carmona-Saez *et al.*, 2006).

The second class of techniques analyzes expression data in the context of additional biological knowledge. Therefore, it represents so-called supervised methods and is also called class comparison (see Allison *et al.*, 2006). One approach is the exploitation of biomolecular networks or gene classification schemes. The latter are often represented as ontologies (see Zareparsi *et al.*, 2004). The standard functional classification system is the Gene Ontology (chapter 1.3.2) which was also applied for the knowledge-based analysis of microarray gene expression data (see Khatri and Draghici, 2005; Zhang *et al.*, 2005). This GO-based type of high-level analysis is the focus of the remaining chapters, since it is also the main topic of this thesis.

Motivation for the Gene Ontology-based high-level analysis

After preprocessing DNA microarray-based gene expression profiling raw data and the identification of differentially expressed genes, useful information on the expression levels of hundreds or thousands of genes, e.g. all genes of a particular bacterium, is obtained. An important follow-up task is the biological interpretation of these data in a pairwise comparison of two or more experimental conditions of interest exploiting collective properties of groups of genes, since individual genes do not operate independently within the cell but in concert (taken from Dopazo, 2006). Therefore, such an analysis includes, for example, the identification of cellular functions or biochemical pathways, whose genes differ strongly in their gene expression profile between the two experimental conditions (see also Khatri and Draghici, 2005; Zhang *et al.*, 2005). This includes the screening of long lists of interesting genes, e.g. differentially expressed genes, or even whole expression matrices. Due to the huge amount of expression data obtained, this is a challenging procedure, since in order to determine the biological role of each gene, information from the literature or public databases has to be collected and taken into account. Performing this in a manual gene-by-gene analysis is also very time-consuming. Nevertheless, this technique was applied to a subset of genes by many researchers. However, it is difficult to obtain a comprehensive overview on the most relevant functions for hundreds of genes in this way and regulatory principles of general nature are often missed. Thus, an automatic evaluation of the genes' available functional information was desirable. For this purpose a sophisticated functional classification system of genes is required such as the GO, since it is based on a solid well-structured ontology and is applicable for every organism (see chapter 1.3.2). In the meantime several tools have been developed that take advantage of the GO classification system for an automated analysis of expression data (see Tab. 4).

An illustrative example for a successful application of such an analysis is the study of McCarroll *et al.* (2004). The authors investigated the process of aging in two distantly related organisms, the nematode *Caenorhabditis elegans* and the fruitfly *Drosophila melanogaster* by using Affymetrix oligonucleotide arrays. Despite the fact that most of the differentially expressed genes were specific to worms or flies, the analysis revealed

Table 3: 2 × 2 contingency table, which applies to the hypergeometric and Fisher's exact test. It shows the binarization of genes into *interestingly expressed* and remaining ones. N is the total number of (measured) genes on the microarray and n is the total number of interesting genes. Analogously, K represents the number of all genes of the current GO node and k the number of interesting genes in the GO node. The nomenclature is the same as in Equation 6.

	genes of interest	remaining genes	$\sum$
In category	k	K - k	K
Not in category	n - k	(N - n) - (K - k)	N - K
$\sum$	n	N -n	N

shared GO categories that were affected. They involved ATP-dependent cellular transporter, mitochondrial metabolism, DNA repair and peptidolysis (McCarroll *et al.*, 2004).

Methodical foundations and used statistical tests
In the following, the algorithmic and statistical foundations for the detection of the relevant GO nodes are briefly reviewed. The simplified workflow of a typical analysis is summarized in the following (see also Fig. 13).

1. Import of the gene expression data and recognition of the gene names
2. Assignment of the genes and, where appropriate, their expression values to the GO nodes
3. Performance of the statistical tests to compute a p-value for each GO node
4. Correction for the multiple testing effect
5. Representation of the results

The most critical step on the outcome of the analysis is the statistical test applied (step 3). Therefore, the different types of tests used by the methods that were published so far are presented below. Basically, two types of statistical approaches exist for the identification of interesting GO nodes: threshold-based and threshold-free approaches (see also Dopazo, 2006). The first one requires a preselection of the genes which are regarded as interesting. This selection is made on the basis of a chosen threshold value and comprises genes marked as differentially expressed according to a particular level of significance, e.g. using 0.05 as threshold value, or genes that were found to be up- or down-regulated by more than a certain factor, e.g. taking two-fold as cut-off value (see also Allison *et al.*, 2006; Dopazo, 2006). The main representatives of this group are the hypergeometric test and Fisher's exact test (see below). The second type of approaches does not need a predefined threshold value to pick out genes. It rather takes the continuous nature of gene expression into account and uses all consistently measured expression values, such as expression ratios or pde (see also Allison *et al.*, 2006; Dopazo, 2006). Three tests that were applied in this context are the t-test, the KS-test and the U-test (see below).

Hypergeometric and Fisher's exact test:
Both, hypergeometric and Fisher's exact test, are based on the same discrete probability

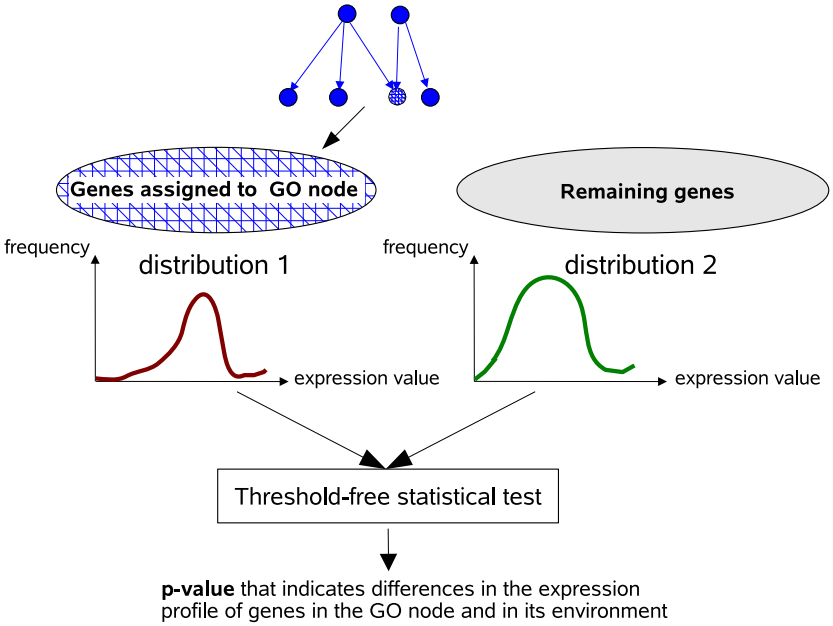


Figure 5: Principle approach of a threshold free statistical test.

distribution, the hypergeometric distribution. The latter models the amount of successes without replacement in a sequence of n drawing from a finite sample. Since the tests are based on a threshold value, a binarization of the genes into interesting and non-interesting ones occurs. An analogous dichotomy is consequently also found for each GO node. This circumstance is formalized in a 2×2 contingency table (see Tab. 3). Given the number k of interesting genes assigned to a selected GO node, the probability can be computed for obtaining such a result. The following hypergeometric distribution or hypergeometric test is used employed to determine the corresponding p-value (Eqn. 6):

$$p = \sum_{i=k}^{min(n,K)} \frac{\binom{N-K}{n-i}\binom{K}{i}}{\binom{N}{n}} \tag{6}$$

Here (Eqn. 6), N denotes the total number of measured genes on the array and n the total number of interesting genes on the array. Analogously, K represents the number of all genes of the selected GO node and k – as mentioned above – the number of interesting genes in the GO node (same nomenclature as in Tab. 3). When the proportion of interesting genes of the actual GO is equal to the overall proportion of interesting genes on the whole microarray, it would be computed by $k_e = (n/N)K$ (expected value). If k exceeds this value, the GO node would be enriched with respect to the number of interesting genes. The probability p for such an enrichment and any more extreme outcomes (see iterating index i in Eqn. 6 which starts at k) can be computed according to the hypergeometric test (see e.g. Zöfel, 2002; Zhang *et al.*, 2004). In the one-sided form, Fisher's exact test yields the same results and thus is identical to the hypergeometric test. If Fisher's exact test is performed with a two-sided alternative hypothesis, in addition to the enrichment of k also its depletions are incorporated that are at least as unlikely. In contrast to the χ^2-test, the hypergeometric and Fisher's exact test are also applicable for small frequencies (see Zöfel, 2002).

Student's t-test:

The Student's t-test is based on a continuous probability distribution, the Student's t-distribution. Since it is threshold-free, no preselection of genes occurs. The expression value distribution of the genes belonging to a particular GO node is compared to the background distribution, e.g. of all genes that do not belong to the node. A similar approach is also valid for the other two threshold-free methods, the KS-test and the U-test, and is outlined in Fig. 5. The parameter that is used for the comparison of the two distributions in the t-test is the arithmetic mean. Thus, the two-sided null hypothesis is that the mean values of the two samples are equal (see Zöfel, 2002). Additionally, the variances of both distributions are considered for the computation of the test statistic (see Eqn. 7). The following equation describes the computation of the test statistic $\widehat{t}$, whereas $\overline{X}$ and $\overline{Y}$ are the arithmetic means, S_X^2 and S_Y^2 the variances, and n_X and n_Y the sizes of the two samples X and Y:

$$\widehat{t} = \frac{\overline{X} - \overline{Y}}{\sqrt{\frac{S_X^2}{n_X} + \frac{S_Y^2}{n_Y}}} \tag{7}$$

This equation applies to samples or distributions with unequal variances (heteroscedastic t-test), which is expected to be appropriate for almost each GO node since the two distributions are normally different in size. The test statistic $\widehat{t}$ allows to determine the corresponding p-value with the help of the Student's t-distribution. Generally, the Student's t-test requires input data that fit a normal-like distribution to get a high power (see Zöfel, 2002).

Kolmogorov-Smirnov test:
The Kolmogorov-Smirnov test is also threshold-free and, therefore, the whole expression value distributions of the genes of a GO node and its background distribution are compared (see Student's t-test and Fig. 5). The input distributions are of continuous nature and no assumptions such as a Gaussian distribution shape are required. For this reasons, it is also called a parameter-free test (see Zöfel, 2002). The cumulative distribution functions are computed in the first step and, if necessary, normalized by the sample size. Then, the maximum vertical deviation D between the two resulting curves is determined (Fig. 6). In contrast to the t-test, where the test statistic is affected by a scaling transformation such as the logarithm, this is not the case for the D test statistic. Lastly, using the obtained D and the Kolmogorov-Smirnov distribution, a p-value can be computed.

Unpaired Wilcoxon's test:
The unpaired Wilcoxon's test – also known as Mann-Whitney U-test – is a rank-based procedure, which answers the question, whether the medians of two independent samples are significantly different. Like the Student's t- and the Kolmogorov-Smirnov test, the unpaired Wilcoxon's test is threshold-free. In addition, like the Kolmogorov-Smirnov test it is parameter-free, thus no assumption on the shapes of the empirical distribution are made (see Zöfel, 2002). Firstly, the values of the two samples are pooled and sorted according to their sizes. Then, the ranks are determined and the rank sums R_1 and R_2 of both samples X and Y are computed as well as the parameters U_1 and U_2, the latter according to the following equations (see Köhler *et al.*, 2002):

$$U_1 = n_1 \times n2 + \frac{n_1 \times (n_1 + 1)}{2} - R_1 \tag{8}$$

$$U_2 = n_1 \times n2 + \frac{n_2 \times (n_2 + 1)}{2} - R_2 \tag{9}$$

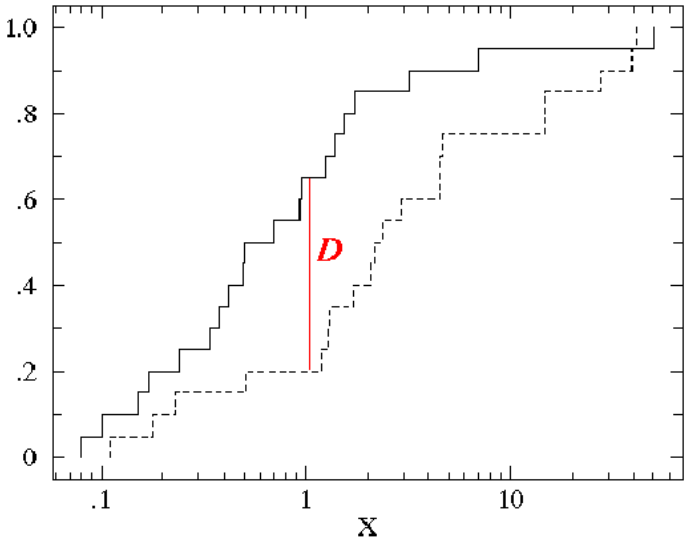


Figure 6: Determining the largest distance D between two cumulative distribution functions, which constitutes the main step in the Kolmogorov-Smirnov test. The distributions in this example are in the logarithmic scale. The figure was taken from http://www.physics.csbsju.edu/stats/KS-test.html.

Here (Eqn. 8 and 9), n_1 and n_2 represent the sizes of the samples X and Y. For the computation of the p-value the minor of the two vales U_1 and U_2 is determined, which is used for looking up the U-distribution.

Finally, it should be mentioned that all described threshold-free tests are used in their independent version, since the two underlying samples, genes within a GO node and the remaining genes, are also independent of each other.

After choosing a particular statistical test, this test is separately applied to each GO node, to which enough genes are assigned. Therefore, their p-values have to be adjusted due to the multiple testing effect (see above). There are two commonly used principal approaches for the correction of this effect: control of the familywise error-rate (FWER) and control of the false discovery rate (FDR). Traditional approaches concern the FWER, which correspond to controlling the α error and, thus, the probability of wrongly rejecting a true null hypothesis. One popular representative is the Bonferroni correction (Bonferroni, 1936, see also Bland and Altman, 1995). With these the probability of erroneously rejecting even one of the true null hypotheses, e.g. for an α error of 0.05, is critical (see Benjamini and Yekutieli, 2001). Therefore, these approaches are comparatively conservative, which means that they have a high risk error of accepting wrong null hypothesis (β) and, for example, miss many true differences of the means or medians between two distributions (see Köhler *et al.*, 2002). A more recently developed approach is the control of the FDR, which is the expected proportion of erroneous among all rejections (Benjamini and Hochberg, 1995; Benjamini and Yekutieli, 2001). It is a more robust method that can handle more errors, when numerous hypotheses are rejected and less when fewer are rejected (Benjamini and Yekutieli, 2001). Thus, the advantage of the FDR method compared to FWER methods, is its greater statistical power. The applied methods for correction of the multiple testing effect are revisited in the context of the existing tools (see below).

Existing tools and their features:
The development of tools and algorithms for the (GO-based) functional interpretation of microarray expression data is an active field of research. Consequently several tools exist that often implement one of the statistical tests presented above, sometimes also in a modified form (e.g. the GSEA method, which is a variant of the KS-test). An assortment of them and their features is shown in Table 4. The main distinctive feature is the statistical algorithm for the identification of the interesting GO nodes (see above). While the precursor of most tools (Doniger *et al.*, 2003) even did not offer a statistical test at all, but only a normalized score, the so-called z-score, the next generation of programs employed threshold-based statistical tests. Therefore, most of the elder tools often provide either an implementation of the hypergeometric or Fisher's exact test (Zeeberg *et al.*, 2003; Al-Shahrour *et al.*, 2004; Martin *et al.*, 2004; Zhang *et al.*, 2004). Both tests are identical, when Fisher's exact test is performed with a one-sided alternative hypothesis (see chapter 2.7.4). In addition, with the χ^2-test an analogous threshold-based method was used in this context, too (see Zhong *et al.*, 2004). Shortly after the creation of these tools, the introduction of the first threshold-free algorithms and their implementations followed. Presumably the predecessor in this field was the algorithm of Mootha *et al.* (2003), which is related to a Kolmogorov-Smirnov test, and its implementation (Subramanian *et al.*, 2005). Other tools that implement one of the threshold-free tests were created by Boorsma *et al.* (2005) (Student's t-test), Ben-Shaul *et al.* (2005) (Kolmogorov-Smirnov test) and (Barry *et al.*, 2005) (Wilcoxon rank sum test). At the same time, some permutation-based approaches were developed (e.g. Volinia *et al.*, 2004).

With respect to the method applied for correcting the multiple testing effect, several early tools do not perform such a correction step at all (Zeeberg *et al.*, 2003; Zhang *et al.*, 2004; Zhong *et al.*, 2004). The other programs offer FWER or FDR methods or both. The presentation of the results comprises table-like lists of GO nodes with their p-values, tree views of GO and sometimes DAG-based visualizations of the relevant parts of the GO graph; different capabilities are provided for exporting and storing these outputs. In addition to GO nodes, some tools also allow the use of other biological groupings (see chapter 1.3.2). For example, the web-based program of Boorsma *et al.* (2005) supports groups of genes bound by a common transcription factor based on ChIP-chip data.

Altogether, most of the tools offer a single threshold-based or threshold-free statistical algorithm together with a particular correction method for the multiple testing effect and often static visualization capabilities. Furthermore, they support the direct analysis of expression data derived from classical eukaryotic model organisms such as man, mouse, thale cress and yeast. Data from prokaryotic organisms are normally not supported or manual effort has to be performed for their inclusion, and no tool existed for the comparative analyses of the most important threshold-based and threshold-free algorithms. In order to fill this gap, an integrative tool should be established that provides several different statistical methods and allows the straightforward analysis of expression data from prokaryotes (see chapter 1.5). In addition, it should offer appropriate methods for correcting the multiple testing effect, an enhanced recognition of alternative names of prokaryotic genes and powerful interactive visualization capabilities, including both a tabular and a subgraph view of GO. With the establishment of the JProGO tool, a step towards this direction was made (Scheer *et al.*, 2006).

Table 4: Existing tools for the functional interpretation of microarray gene expression data (Status: spring 2007). Due to the lack of space only a fraction of all available tools is shown. For a comprehensive review on this topic please refer to Nam and Kim (2008). It should be mentioned that the hypergeometric test is identical to a Fisher's exact test with one-tailed alternative hypothesis (see chapter 2.7.4).

Name & Reference	**Statistical analysis**	**Multiple testing correction**	**Implemented in**	**Access**
Catmap (Breslin *et al.*, 2004)	Fisher's exact test, Wilcoxon's test	yes	Perl	local program
FatiGO (Al-Shahrour *et al.*, 2004)	Fisher's exact test	yes	?	web interface
GOAL (Volinia *et al.*, 2004)	permutation-based	yes	Perl	web interface
GOdist (Ben-Shaul *et al.*, 2005)	Fisher's exact test, Kolmogorov-Smirnov test	yes	Matlab	local install.
GO-Mapper (Smid and Dorssers, 2004)	expression quotient	—	Perl	local install.
GoMiner (Zeeberg *et al.*, 2003)	Fisher's exact test	no	*Java*	local install.
GOSurfer (Zhong *et al.*, 2004)	chi-squared test	no	? (Windows®-based)	local install.
GO:TermFinder (Boyle *et al.*, 2004)	hypergeometric test	yes	Perl	local install.
GOTM (Zhang *et al.*, 2004)	hypergeometric test	no	*PHP*	web interface
GOTool Box (Martin *et al.*, 2004)	hypergeometric test, binomial distribution	yes	Perl	web interface
GSEA (Gene Set Enrichment Analysis) (Subramanian *et al.*, 2005)	GSEA method (related to Kolmogorov-Smirnov test)	yes	*Java* / R	local install.
JProGO (Scheer *et al.*, 2006 and this work)	Fisher's exact test, Student's t-test, Kolmogorov-Smirnov test, Wilcoxon's test	yes	*Java* and R	web interface
MAPPfinder (Doniger *et al.*, 2003)	z-score calculation	no	?	local install.
Significance Analysis using Structured Permutations (Barry *et al.*, 2005)	permutation-based	yes	R	local install.
T-profiler (Boorsma *et al.*, 2005)	Student's t-test	yes	?	web interface

1.5 Objectives of this Work

The invention of high-throughput technologies in transcriptome research, especially the microarray technology, led to an explosion of the available gene expression data. Several tools exist for the automatic interpretation of such large scale gene expression data using classification systems such as the Gene Ontology (GO). However, most of these tools only focus on expression data from eukaryotes and offer a single statistical algorithm for the detection of interesting GO terms. Therefore, in the first part of this work a novel program suite for the functional interpretation of DNA microarray data (JProGO) had to be developed, whose focus lies on expression data from prokaryotes. In addition, several different algorithms for the identification of GO terms with significantly altered gene expression profile should be included, comprising both threshold value-based and threshold value-independent methods. Further features of the tool had to be the support of different types of expression data, the recognition of alternative gene names, an appropriate correction for the multiple testing effect and an intuitive and interactive visualization of the obtained results, taking the directed acyclic graph structure of GO into account. In conjunction with the implementation of the JProGO program, the PRODORIC database should be expanded to include the functional annotations of GO.

In the second part a case study on bacterial expression data had to be performed in order to test the developed JProGO tool. In this context, the different statistical algorithms should be compared and the influence of the type of expression data – expression ratios versus probabilities of differential expression – should be evaluated.

Thirdly, JProGO was to be applied for the analysis of microarray expression data from in-house experiments, mainly from the pathogenic bacterium *Pseudomonas aeruginosa*. For this purpose, a combined low- and high-throughput analysis had to be performed. In the low-level analysis different preprocessing algorithms were to be compared and the outcome on the high-level interpretation using JProGO was to be evaluated. Furthermore, the relevant GO nodes should be identified and critically discussed.

Finally, the JProGO approach should be expanded from GO to other groups of genes, especially to experimentally verified regulons from transcriptional regulatory networks that are stored in the PRODORIC database. For this purpose, appropriate expression data sets from gene knockout strains were to be used.

2 Materials and Methods

2.1 Hardware

Two development computers were used for this thesis: The first one was a Medion MD40100 notebook with an Intel® Pentium 4 Mobile processor with 2.80 GHz clock frequency, 1024 MB RAM and a 100 GB hard disc drive. The second development computer, which completely replaced the first one, was an Asus Z 92 series notebook containing a Core Duo T2400 Centrino CPU chip set from Intel® with a clock frequency of 1.83 GHz. The RAM capacity was 2048 MB and that of the hard disc 120 GB. For hosting the JProGO web-based program suite, a desktop computer (vertical tower) with a Core 2 Duo 6600 System CPU chip set from Intel® (2.40GHz clock frequency) was used. The allocated RAM was 3.1 GB and the capacity of the hard disc 140 GB.

2.2 Operating Systems

As the operating system throughout this work several distributions of Linux (32 bit version) were used. The SUSE Linux versions 9.0 and 9.3 (kernel 2.6.11.4), respectively, were employed for the first development computer. SUSE Linux version 10.1 (kernel 2.6.16.21) was installed on the second development computer (see chapter 2.1). The computer that was employed as the web server (chapter 2.1) was run under Ubuntu Linux version 7.04 (version name: 'Feisty Fawn'), whereas the used kernel version was 2.6.20-16 (generic).

2.3 Programming Languages, Libraries and Extensions

2.3.1 *Java*

Java is an object-oriented, platform independent programming language developed by Sun Microsystems (`http://java.sun.com`). It was employed as the principal programming language throughout all software projects, especially for the development of the JProGO program (chapter 2.7). In this context, it was made use of the consequent object-oriented design of *Java*, which allowed for the creation of reusable classes and well-defined interfaces. *Java* developmental kit (JDK) versions 1.4 and 1.5 were chosen, including their standard libraries. Furthermore, several free *Java* class libraries (`http://java.sun.com/j2se/1.4.2/docs/api/`, `http://java.sun.com/j2se/1.5.0/docs/api/`) were used for various areas of application. They are listed in the following:

- Servlet API, version 2.4 (`http://java.sun.com/products/servlet`): allows the development and execution of *Java* web applications using the server-sided Servlet technology
- JRClient, version RE817 (`http://rosuda.org/Rserve`): *Java* client API for sending commands to and obtaining results from *R*, which has to be run in a server mode (see Rserve in chapter2.3.2)
- File Upload, version 1.0 (`http://commons.apache.org/fileupload/`): adds a file upload capability to *Java* web applications, e.g. servlets

- JFreeChart, version 0.9.20 (`http://www.jfree.org/jfreechart`) and JCommon, version 0.9.5 (`http://www.jfree.org/jcommon/`): JFreeChart allows for the creation and display of charts, e.g. X-Y, bar and pie charts; JCommon is a compilation of classes used by JFreeChart
- JDBC driver for PostgreSQL, versions 7.3 and 8.1 (`http://jdbc.postgresql.org/`): type 3 driver for accessing databases run under the PostgreSQL DBMS (chapter 2.4.3)
- Xerces Parser (`http://xerces.apache.org/`): used SAX (Simple API for XML) implementation for the event-driven parsing of XML documents (additional packages: resolver.jar and xml-apis.jar)
- GNU JAXP (`http://www.gnu.org/software/classpathx/jaxp/`): free implementation of the standard XML processing APIs for *Java*, e.g. SAX, DOM, and JAXP (from Sun)
- Common Collections (`http://commons.apache.org/collections`): provides data structures for collections such as queues and expands the existing collection classes from the JDK
- Jutil (`http://org-jutil.sourceforge.net`): multi-purpose API for expanding various functionalities of the *Java* standard class library, e.g. support of regular expressions

2.3.2 *R* and Bioconductor

While *Java* was the basic programming language for all self-developed software projects such as JProGO, the free programming language *R* was employed for statistical computations and visualizations. It was mainly used for the execution of statistical tests within the JProGO program suite (chapter 1.4.1) and for the low- and mid-level analysis of microarray expression data (chapter 1.4.2). Version 2.3.1 of *R* was obtained from `http://cran.r-project.org/`. It was compiled and installed directly from the source code. As for *Java* (chapter 2.3.1), several additional packages were utilized to increase the functionality of *R*. They largely comprise libraries for the preprocessing and mid-level analysis of microarray gene expression data (chapter 2.8 and 2.9), which are part of Bioconductor (Gentleman *et al.*, 2004). Bioconductor is an open source software project, which is released two times per year as an extension to *R* (`http://www.bioconductor.org`). In the following, the regularly applied Bioconductor packages are listed. They were all compatible with *R* version 2.3.1 and were obtained from the official web site (`http://www.bioconductor.org`) using the getBioC *R* script:

- affy: contains various methods for facilitating the handling of data from Affymetrix GeneChip® oligonucleotide arrays
- annotate: provides annotations for microarrays and other metadata
- Biobase: offers the basic functionalities of Bioconductor
- gcrma: allows performing background adjustment taking advantage of sequence information (gcrma method)

- MASS: package for enhanced cluster analysis
- siggenes: performs significance analysis of microarrays
- simpleaffy: wrapper for the affy package (see above) that provides data exploration facilities for the Affymetrix GeneChip® platform
- vsn: enables the execution of variance-stabilized preprocessing of microarray expression data (vsn method)

Further installed *R* libraries and packages, which were not part of the Bioconductor project, were:

- Rserve, version 0.3-17 (`http://rosuda.org/Rserve/`): add-on server program for *R*, which allows responding to requests from other programming languages such as *Java* and *C++* via appropriate clients, e.g. JRclient for *Java* (see chapter 2.3.1)
- CyberT/bayesreg (`http://cybert.ics.uci.edu/cgi-bin/CyberTReg-8.0.form.pl`, see also Baldi and Long, 2001): provides Bayesian-based regularized t-tests and statistical inferences for the detection of differentially expressed genes

2.3.3 Unix Shell Programming

The classical command-line user interface in Unix-like operating systems such as the Linux employed in this work (chapter 2.2) is called shell. Amongst executing single commands, the shell can be used as a full scripting language. The bash (Bourne-Again-Shell) was utilized as shell for the creation of scripts that were mainly applied for parsing and importing GO and GOA into the PRODORIC database (chapter 2.7.2).

2.3.4 SQL

The Structured Query Language (SQL) was employed for accessing and querying databases that were installed on the development computers (chapter 2.1). The main database was PRODORIC (see Münch *et al.*, 2003, 2005 and chapter 1.3.1, 2.5.2) and PostgreSQL was the corresponding DBMS (for versions, see chapter 2.4.3). SQL does not represent a full programming language and in each case needs another (full) programming language or an appropriate database client for its invocation. Thus, SQL commands were embedded in the source code of the *Java* programming language or generated, for example, by Unix shell programs (chapter 2.3.3) and executed by the client program psql (chapter 2.4.3).

2.3.5 *PHP*

PHP (Hypertext Preprocessor) is a scripting language, which is often used for the creation of dynamic web pages. Since *PHP* had been employed before for the web interface of the PRODORIC database by Münch *et al.* (2003), it was also used for displaying the expansions of the PRODORIC database that were made throughout this thesis (inclusion of GO and GOA). Versions 4 and 5 of *PHP* were employed. The adaption of the PRODORIC web sites was mainly done by Richard Münch and Claudia Hundertmark from the Technische Universität Braunschweig (Münch *et al.*, 2005).

2.4 Used Programs and Software

Utilized programs and tools that were developed and implemented by others are listed below:

2.4.1 Integrated Development Environments

In principle, the development of software is possible using solely a text editor and a few other components such as a runtime environment and, if necessary, a compiler. However, integrated development environments (IDE) clearly ease the process of software development by integrating the facilities mentioned above and additional ones. *Java* programming was done with the help of such an IDE, the JBuilder® from Borland (now CodeGear). It contains a source code editor with syntax highlighting, an enhanced debugger, a build automation and compiling tool, a *Java* runtime environment and supports the creation and deployment of *Java* web projects. All these features as well as the archive builder were used for the development of *Java*-based software. The three versions JBuilderX, JBuilder 2005 and JBuilder 2006 were employed.

2.4.2 Web Server Software

Tomcat (`http://tomcat.apache.org/`), which is the official reference implementation for the Java Servlet and Java Server Page technology, was chosen as the servlet container and web server for the JProGO program suite. Version 5.5.9 runs on the web server computer under an Ubuntu Linux platform (chapters 2.1 and 2.2).

The Apache HTTP Server, which is the most commonly used web server supporting a variety of server-sided scripting languages such as *PHP* and *Perl*, was employed for running the extended PRODORIC web interface (mainly work of Richard Münch, see Münch *et al.*, 2005). Version 2.x of Apache HTTP Server was taken.

2.4.3 Database Management Systems

The term database management system (DBMS) refers to the administration software which is – based on the underlying database model – required to run and organize the access to a database, e.g. controlling all reading and writing operations. In this thesis, the object-relational DBMS PostgreSQL (`http://www.postgresql.org`) was utilized for running local copies of the PRODORIC database on the development computers. PostgreSQL Version 8.0.3 was used on the first development computer and version 8.1.4 on the second (see chapter 2.1). The also employed command-line client psql had the same version number as the DBMS.

2.4.4 Sequence Alignment Tools

Throughout this thesis, the basic local alignment search tool (BLAST) was utilized for the pairwise alignment of sequences. BLAST is a commonly used collection of programs for the analysis and comparison of biological sequence data e.g. the nucleotide sequence of DNA molecules or the amino acid sequences of proteins. The BLAST programs are developed by the NCBI and are based on heuristic algorithms for finding (nearly) optimal pairwise local alignments. It was introduced by Altschul *et al.* (1990) and serves

as a fast prefiltering step for the subsequent exact alignment using a dynamic programming approach. A BLAST search allows for comparing a single query sequence with a whole database of sequences. It identifies the sequence or sequences that closely resemble the input sequence providing a solid statistical measure, the E-value, for assessing the similarity.

BLAST is often used as a web-based tool on the NCBI homepage (`http://www.ncbi.nlm.nih.gov/blast/Blast.cgi`) or other web hosts, and typically a single or a few sequences are submitted for such an online query. Since for the thesis at hand a large number of input sequences had to be compared to the same database, usage of the online tool was unfeasible. Therefore, a local version of BLAST was chosen. For this purpose, version 2.2.9 of blastall was downloaded as Linux executable from the NCBI ftp server (`ftp://ftp.ncbi.nih.gov/blast/executables/`) and installed on the development computers. In order to prepare a database, to which the input sequences could be compared, a self-developed *Java* program was employed. With its help a FASTA file was generated from PRODORIC that contains all protein sequences of the genome from which the query sequences were derived (see chapter 2.7.3). With this setup blastp analysis with protein sequences as input sequences (*"formatdb -i FASTA-file -p T"* and *"blastall -p blastp ..."*) were performed. They were invoked by another *Java* program.

2.4.5 Graph Visualization Tools

Through the visualization of graphs their mathematical representation is converted into a human-readable format on a two-dimensional surface. This facilitates the recognition of essential information such as the relationships between different nodes (see also Battista *et al.*, 1994). It belongs to the research field of graph drawing and several layout algorithms have been developed which, for example, minimize the crossing of the edges. In this thesis, Graphviz (`http://www.graphviz.org`), an open source graph visualization software with different layout capabilities, was utilized. It contains the dot tool (Gansner *et al.*, 2002, 2006) that is especially suited for the drawing of directed acyclic graphs such as GO. It is used for the visualization of the analysis results of the JProGO web software (see chapter 2.7.5). Version 2.8 of dot is used.

2.4.6 Miscellaneous Tools

In addition to the programs listed above, the following tools were used:

- DBVisualizer (Minq Software, version 5.0): visualization of the structure of a database by generation of entity relationship-like diagrams (e.g. chapter 2.6)
- Kile (version 1.8.1): integrated LaTeX environment used for writing and typesetting the thesis at hand

2.5 Employed Data Resources and Databases

2.5.1 Microarray Data Sets

Several microarray gene expression data sets derived from the three prokaryotic organisms *Escherichia coli* (strain K12), *Bacillus subtilis* (strain 168) and *Pseudomonas aeruginosa* (PAO1) were analyzed in this work. With respect to the extent of the required analysis efforts, they can be broadly divided into the following two groups:

1. already preprocessed expression data
2. raw expression data

In order to obtain appropriate preprocessed data sets (Item 1), the literature was screened with the help of the PubMed and GEO database (chapter 1.2.4). The found data sets were then derived from the supplementary material of the corresponding publications: Hung *et al.* (2002); Salmon *et al.* (2003, 2005); Kang *et al.* (2005) (for *E. coli*) and Keijser *et al.* (2007) (for *B. subtilis*). As type of expression data both, expression ratios and ppde/p-values were taken. A detailed compilation of the preprocessed data sets with the investigated conditions and the type of expression data is given in Table 6 in the *Results and Discussion* section (chapter 3). All preprocessed data sets were employed for a functional interpretation with JProGO, either performed in this thesis (*E. coli* data, chapters 3.2.1 and 3.2.2) or by Keijser *et al.* (2007) (*B. subtilis* data, chapter 3.2.3).

The raw data sets (Item 2) were derived from in-house experiments. Theses were planned and performed by the cooperation partners Dr. Max Schobert (Institute for Microbiology, Technische Universität Braunschweig, Germany) and his co-workers Dr. Kerstin Schreiber, Beatrice Benkert, Nelli Bös and Sabrina Thoma (unpublished manuscripts of Schreiber et al., 2007 and Benkert et al., 2008). The data sets comprised expression profiling experiments from *P. aeruginosa* (PAO1). Again, more detailed information on the experiments, such as the investigated conditions, are given in the *Results and Discussion* section (Tab. 12). In each case the Affymetrix GeneChip® platform was employed and three replicate measurements were performed for each conditions. The raw expression data sets were preprocessed with Bioconductor (see chapter 2.8) and afterwards a mid-level analysis with the CyberT plugin for *R* was performed to compute the ppde (chapter 2.9). The computed ppde were taken for a subsequent high-level analysis with JProGO to identify the significant GO nodes (chapter 2.10).

In general, if more than 6 digits after the decimal were available, before employing them for a high-level analysis the expression ratios and ppde/p-values were rounded at the 6^{th} digit after the decimal.

2.5.2 PRODORIC Database

The PRODORIC database (see chapter 1.3.1 and Münch *et al.*, 2003, 2005) was expanded to include the GO hierarchy and the respective annotation of the gene products, the GOA (see chapter 2.5.3). For this purpose and for the extraction of the regulons (chapter 2.11), a snapshot of PRODORIC was taken in January 2006. The structural extension of PRODORIC and the import of GO and GOA is described in chapter 2.6.

2.5.3 Gene Ontology (GO) and Gene Ontology Annotation (GOA)

GO was downloaded as a MySQL dump in December 2005 using the termdb file (go_YYYYMM-termdb-data.gz, YYYY specifies the year and MM the month). The then download URL was `http://www.godatabase.org/dev/database/archive/latest/` (URL at the time of writing: `http://archive.geneontology.org/`). The gene association file GOA Uniprot was downloaded from the ftp server of the EBI in November 2005 (`ftp://ftp.ebi.ac.uk/pub/databases/GO/goa/UNIPROT/gene_association.goa_uniprot.gz`).

2.5.4 UniProt Database and Genome Reviews

The UniProtKB database (chapter 1.3.1), more precisely their taxonomic division for prokaryotes (uniprot_sprot_archaea.dat, uniprot_sprot_bacteria.dat, uniprot_trembl_archaea.dat, uniprot_trembl_bacteria.dat), was downloaded in the flatfile format in December 2005 (now URL: `ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase`).

For each completely sequenced prokaryotic organism and selected eukaryotes, a Genome Review file exists which contains an annotated genomic sequence of its replicons including a list of the encoded proteins. All Genome Review files representing the prokaryotic organisms of JProGO were downloaded in the DAT format in December 2005 (now URL: `ftp://ftp.ebi.ac.uk/pub/databases/genome_reviews/dat/cellular`). They were concatenated into one file. Afterwards the genomes of interest were extracted from this single file with the help of their NCBI Taxonomy IDs (chapter 2.6).

2.6 Expansion of PRODORIC

2.6.1 Structural Extension of the PRODORIC Database and Import of GO and GOA

Several new relations were created for mapping the information contained in GO and GOA to the initial PRODORIC database. These comprised the three central tables go, go2go and go2polypeptide (Fig. 7) as well as the graph_path table. The first two mentioned relations were based on a similar structure as the corresponding tables of the GO database: The relation go was needed to represent the nodes of the GO graph including the name, accession number and a detailed definition of each GO node. The go2go table was required for storing the directed edges between the GO nodes, which was achieved by using the columns parent_go_no and go_no for representing the parent-child relationships (Fig. 7). The go2polypeptide table was designed for linking the gene products, in particular the polypeptides of the PRODORIC database, to the nodes of the GO hierarchy and, therefore, is referencing the primary keys of both, the already existing polypeptide and the newly created go table. Finally, the graph_path table from the GO database was included, to have a denormalized representation of the GO hierarchy, in which each GO node is not only connected to its direct child nodes as in the go2go table, but with all its successor nodes. This information was required for the computation of the significant GO nodes in the JProGO software suite (chapter 2.7).

After creation of the required database relations, the next step comprised the import of the information from GO and GOA into the extended PRODORIC database. For this purpose, current releases of GO and GOA (text file) were downloaded from the respective web sites (see chapter 2.5.3). A shell script was created that automatically performs the tasks of downloading and importing the data. For the go, go2go and graph_path table the MySQL table dumps were parsed and adapted for compatibility with PostgresSQL. MySQL specific features were eliminated with the help of regular expressions (grep and sed tool). Then, temporary tables were used to insert the data from the adapted SQL dump into the new PRODORIC tables. Since the structure of these relations from PRODORIC did not exactly match to those of the GO database, some adaptations had to be performed: for example, the GO terms and their definitions – two separate tables in the GO databases – were merged into the go table. In order to import the assignments of the polypeptides from PRODORIC (polypeptide table) to the GO terms (go table),

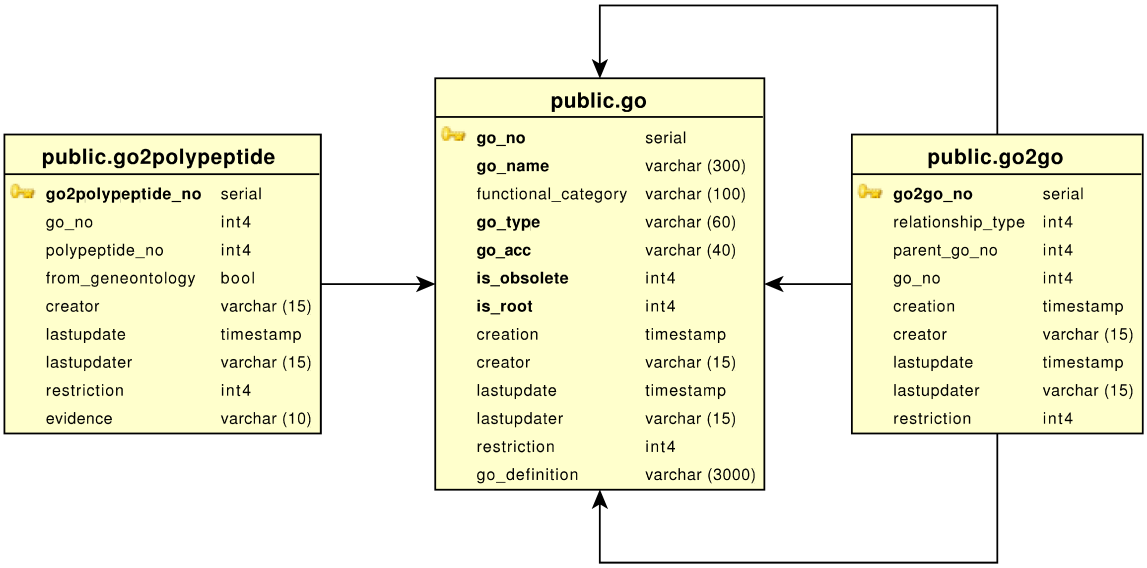


Figure 7: Newly created relations for integrating the Gene Ontology and Gene Ontology Annotation into the PRODORIC database. Database tables are symbolized by rectangles and their 1:n relationships by arcs, whereas an arrow points towards the table which holds the referenced primary key.

information of the GOA flat file and PRODORIC was exploited: a preliminary table was employed for reading in the GOA flat file. Then, in a multi-table SQL statement for filling the go2polypeptide table, on the one hand the stored GO accession numbers were matched with the go table and on the other hand the db_object_id (UniProt accession number) of GOA with those of PRODORIC joining the tables dblink2table, dblink, db.

2.6.2 Upper Level Gene Ontology Categories for the PRODORIC Web Interface

One search option of the gene form of the PRODORIC web interface should be finding genes according to the broad functional groups they belong to (Fig. 11). For this purpose, the nodes of the *Molecular Function* and *Biological Process* branch of GO (table go), the go2polypeptide and the graph_path table were utilized. Three additional tables, class and the two linking tables polypeptide2class and gene2class, were created. The class table was used to store only *Molecular Function* nodes from the third level of the GO hierarchy and *Biological Process* nodes from the fourth level. For this purpose, and for filling the linking tables, an SQL statement joining the go2go and graph_path table was employed (see Appendices). The adaption of the PRODORIC web forms for displaying the GO data was primarily performed by Richard Münch (Institute for Microbiology, Technische Universität Braunschweig, Germany).

2.7 Development and Running of the JProGO Program Suite

2.7.1 Overview on the Development

In the following, the individual development steps of the JProGO software are listed. Their sequence broadly reflects the chronological order as well as the workflow flow of the JProGO program:

1. Conversion of GO and GOA information from the PostgreSQL database PRODORIC to *Java*-based graph objects
2. Database-independent storage and reload of the *Java*-based GO graphs and their organism-specific gene annotations
3. Matching of the preprocessed gene expression data from a microarray experiment to the GO graph
4. Statistical testing of the nodes of the GO graph and correction for the multiple testing effect
5. Interactive visualization of the results

The process of converting the data from the expanded PRODORIC database (step 1), originally derived from GO and GOA, is described in detail in chapter 2.7.2. Here, also the approach and the classes for the object-oriented representation of the GO graph and the gene annotations (GOA) are specified. These graphs have to be stored as files, in order to allow their fast restoring (step 2) for subsequent analysis requests to the JProGO program. The XML-based methods used for this purpose are explained in the second part of chapter 2.7.2. To perform a functional analysis of microarray expression data, the genes have to be mapped to the nodes of the object-oriented GO graph instances – together with their expression levels. The preparatory steps for that, including the extraction of synonyms, and the matching procedure itself is outlined in chapter 2.7.3. Subsequently, the statistical testing of the individual GO nodes with respect to differences in their gene expression profile compared to the background distributions is performed, which is delineated in chapter 2.7.4. In the same chapter the correction for the multiple testing effect is described. The implementation of an interactive representation of the analysis results is summarized in chapter 2.7.5. Finally, in chapter 2.7.6 the implementation of the web frontend is explained.

2.7.2 Import of GO Graphs from PRODORIC and Object-oriented Representation

Readout of the GO and GOA data from the PRODORIC database and their object-oriented representation

Initially, in order to make the GO graphs and the associated genes from the extended PRODORIC database available to the *Java*-based program suite, a database connection to PRODORIC was established using the JDBC technology (type 3 driver, see chapter 2.3.1). Then, appropriate SQL statements were executed to obtain the relevant data, e.g.

```
SELECT DISTINCT parent_go_no , go_no FROM go2go ...
ORDER BY parent_go_no , go_no

SELECT DISTINCT go_no , parent_go_no FROM go2go...
ORDER BY go_no , parent_go_no
```

for the identification of all parent-child as well as child-parent relationships of the GO nodes. In addition, all successors of each GO node were determined using the graph_path table and the assignments between the gene products and their GO nodes (joining the tables go, go2polypeptide, polypeptide, gene, gene2replicon, replicon, genome) were established. All data were hold in a non-rectangular two-dimensional array, a special kind of data structure which is organized analogously to the adjacency list often employed in graph theory. These arrays served as an in-memory interlayer and two self-created central classes Graph and Node were used to generate graph objects from them. This was done for each of the more than twenty prokaryotic organisms (see Tab. 5), which are directly supported by JProGO. The graph structure was the same for all organisms, but due to their different genomes the genes that were annotated to the GO nodes differ. To save memory, the gene products assigned to each GO node were represented by an array of bits. The position in the array specifies a particular polypeptide and a set bit reflects its assignment to the GO node. Therefore, each Node instance contains a BitArray object (see Fig. 8). Furthermore, the Node class provides access methods for obtaining a node's child and parent nodes. This corresponds to an implicit adjacency list representation considering, for the sake of increased performance, both directions, child-to-parent as well as parent-to-child relationships of each node. Since these relationships, the edges, were all modeled indirectly by the Node objects themselves, no extra Edge class was necessary. In addition, the Node class offers, through the implementation of the IGoNode interface, further useful access methods. They allow, for example, to determine all successors of a node, the genes directly assigned to it as well as those genes that are also linked to all of its successor nodes. For the two latter purposes the above mentioned BitArray class is employed (Fig. 8). These methods and the method setAnalyzerResults (class AnalyzerResult) were required for the subsequent statistical evaluation of each GO node (chapter 2.7.4) and the subgraph-based visualization of the results (chapter 2.7.5). Amongst the Node class, the second central class for storing a GO graph was the Graph class, which implements the interface IGoGraph (Fig. 8). An instance of this class was employed to store all nodes (Node instances) belonging to the same graph. It can return a list of these nodes as well and the number of genes representing an organism-specific GO graph. Further methods provide fast access to the annotations of the nodes and their assigned gene products, e.g. getting the node's by its number or getting a polypeptide's name by its position in the BitArray instance (Fig. 8). This *Java*-based representation of organism-specific GO graphs was the basis for all subsequent steps of analysis in the JProGO software (chapters 2.7.4 and 2.7.5).

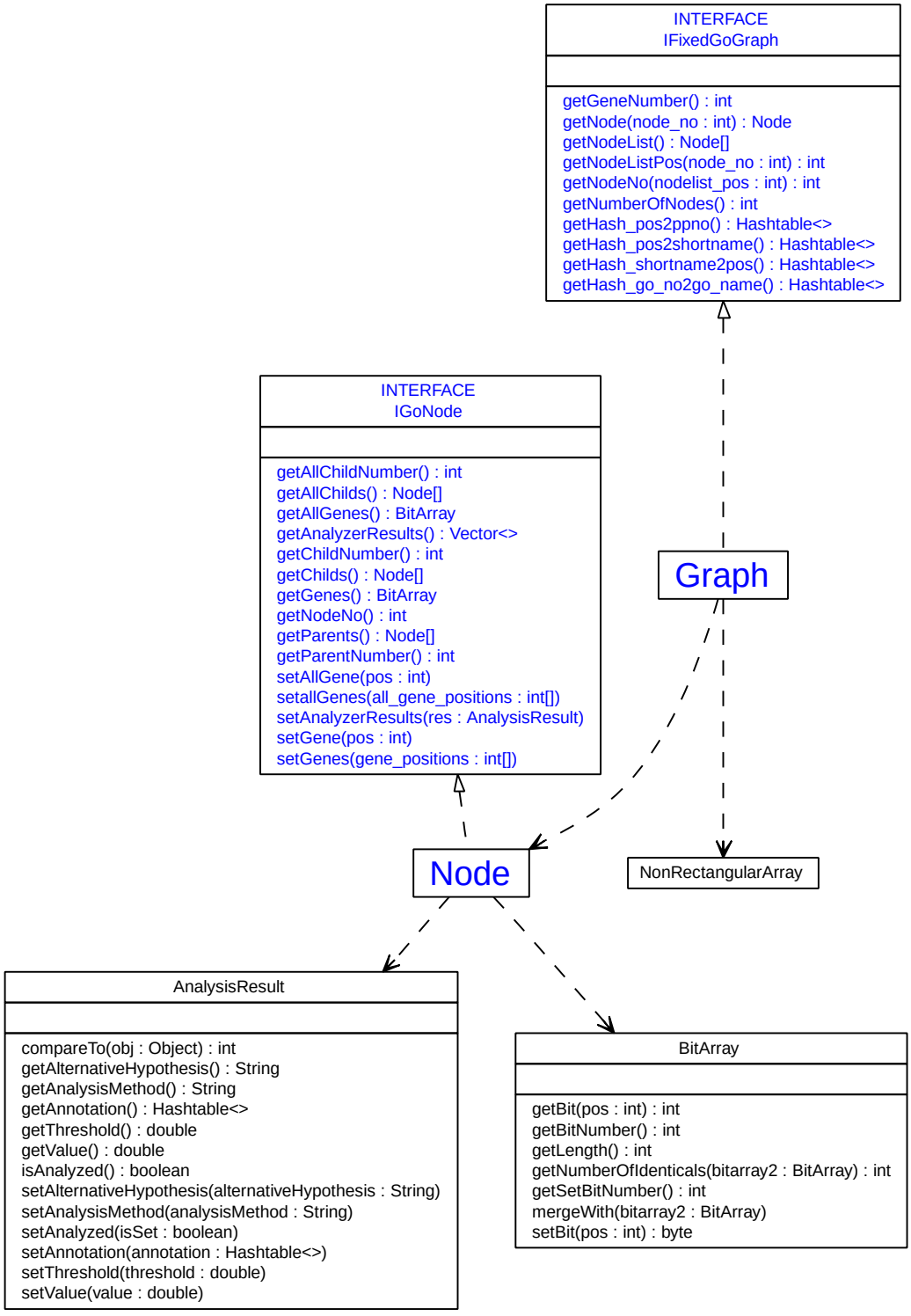


Figure 8: UML-like diagram of the classes used for the *Java*-based representation of the GO graph and the genes assigned to it. The central classes are Graph and Node (blue font) which implement the interfaces IGoNode and IFixedGoGraph, respectively. Hollow triangles symbolize the inheritance relationship. Selected methods of the two interfaces and of the classes Analysis Result and BitArray are shown. The arrows represent composition relationships that point from the container class to the dependent class. The diagram was constructed with the UMLGraph tool and the graphviz program (chapters 2.4.5 and 2.4.6).

Table 5: The 23 prokaryotic species directly supported by JProGO together with the number of genes. Solely chromosomal protein encoding genes were considered and the data were imported from the PRODORIC database (status: January 2006). The table was taken from the original publication about JProGO (Scheer *et al.*, 2006).

Organism	Gene number
Bacillus cereus (strain ATCC 10987)	5603
Bacillus subtilis (strain 168)	4101
Caulobacter crescentus (strain CB15)	3737
Clostridium tetani (strain Massachusetts)	2373
Corynebacterium glutamicum (strain DSM 20300 [Nakagawa])	3099
Escherichia coli (strain K12)	4291
Helicobacter pylori (strain ATCC 700392)	1580
Listeria innocua (serovar 6a, strain CLIP 11262)	2981
Listeria monocytogenes (serovar 1/2a, strain EGD-e)	2855
Methanococcus jannaschii (strain JAL-1)	1770
Mycobacterium tuberculosis (strain H37Rv)	3918
Mycobacterium tuberculosis (strain Oshkosh)	4187
Mycoplasma genitalium (G-37)	480
Mycoplasma pneumoniae (strain M129)	688
Pseudomonas aeruginosa (strain PAO1)	5573
Pseudomonas putida (strain KT2440)	5351
Rickettsia conorii (strain Malish 7)	1374
Rickettsia prowazekii (strain Madrid E)	834
Salmonella typhimurium (strain ATCC 700720)	4412
Staphylococcus aureus (strain N315)	2595
Streptococcus pneumoniae (strain TIGR4)	2094
Streptococcus pyogenes (serovar M1, strain SF370)	1697
Yersinia pestis (biovar Mediaevalis, strain KIM5)	4090

Persistence of the *Java*-based GO graphs and their organism-specific gene annotations

The execution of complex SQL queries can take a considerable amount of time. For example, the construction of the GO graphs from the PostgreSQL database PRODORIC took up to several minutes. Since this process had to be repeated for each JProGO analysis, it would slow down the interactive usage of the program markedly. In order to raise the speed for the *Java*-based reconstruction of the GO graphs (chapter 2.7.2), a text file representation was chosen as storage form. As format well-formed XML was taken for an ordered access to the data and a corresponding class XML Writer was utilized for saving a GO graph instance as XML. For the reconstruction of the graphs another class, XML Reader, was implemented, which takes advantage of the event-driven SaxParser technology (see chapter 2.3.1). The structure of the XML files is described in Fig. 9 exemplarily for *E. coli* (strain K12). In brief, the root element is named graph ($< graph > ... < /graph >$), and it contains the three elements *genepos2polypeptide_list*, *genepos2shortname_list* and *gonode_list*. The first two elements store the polypeptide annotations, in particular the assignment of the polypeptides' positions in the BitArray to their PRODORIC accession numbers and short names, respectively (see interface IFixedGoGraph in Fig. 8). The third element, *gonode_list*, is nested itself and contains a collection of *go_node* elements. These represent the individual nodes and edges (nested element $< child\_list >$) of the graph as well as the polypeptides assigned to the nodes ($< gene\_list >$, not shown in Fig. 9). In addition, as described in chapter 2.7.2, not only the direct children of a node are specified but all successor nodes ($< all\_child\_list >$) and their polypeptides. This yields a list of all gene products that are logically represented by the chosen node ($< all\_gene\_list >$). The latter is needed for the subsequent statistical evaluation of the GO nodes with respect to their gene expression levels (chapter 2.7.4).

Altogether, the reconstruction of a GO graph from the described well-formed XML file with the self-created XML reader class takes only a few seconds.

2.7.3 Matching of Gene Names and Synonyms

An essential prerequisite for the functional interpretation of high-throughput gene expression data using GO is the matching of gene names and identifiers derived from the microarrays to the gene annotations of the nodes in the GO graph (see chapters 1.4.3 and 3.1.2). Therefore, one goal of JProGO was to ensure the recognition of a considerable proportion of genes represented on the microarray to be analyzed and interpreted. A two-step approach was chosen for this purpose:

1. Creation of a compilation of gene names, ordered locus names (OLN), alternative names and synonyms (for each organism)

2. Priority-based matching of the genes from the microarray to the GO annotation using the compilation from the previous step

For the first step – the generation of the compilation of gene names and synonyms – three data sources were employed as input: UniProtKB and Genome Reviews (chapter 2.5.4) as well as the PRODORIC database. An object-oriented in-memory representation of the proteins' short names, OLNs and synonyms (UniProtKB and PRODORIC), the UniProt accession number (UniProtKB and Genome Reviews) and the corresponding protein sequences (Genome Reviews and PRODORIC) was generated. Then, the central

```
<graph>
        <genepos2polypeptide_list>
                <assign pos="0" polypeptide_no="104780" />
                <assign pos="1" polypeptide_no="102867" />
                ...
                <assign pos="4290" polypeptide_no="103827" />
        </genepos2polypeptide_list>
        <genepos2shortname_list>
                <assign pos="0" shortname="Aas" />
                <assign pos="1" shortname="Aat" />
                ...
                <assign pos="4290" shortname="Zwf" />
        </genepos2shortname_list>
        <goname_list>
                <assign go_no="1" name="all" />
                <assign go_no="2" name="is_a" />
                ...
                <assign go_no="20080" name="fructosyl-amino acid oxidase activity" />
        </goname_list>
        <gonode_list>
                <go_node number="1" name="all" child_amount="6" parent_amount="0"
                         all_child_amount="20073" gene_amount="0" all_gene_amount="3438">
                        <child_list>
                                <child no="10" />
                                <child no="27" />
                                ...
                                <child no="4809" />
                        </child_list>
                        <all_child_list>
                                <all_child no="1" />
                                <all_child no="4" />
                                ...
                                <all_child no="20080" />
                        </all_child_list>
                        <all_gene_list>
                                <all_gene no="1" />
                                <all_gene no="2" />
                                ...
                                <all_gene no="4290" />
                        </all_gene_list>
                </go_node>
                <go_node>
                        ...
                </go_node>
                ...
        </gonode_list>
</graph>
```

Figure 9: Self-created XML format for persistent storage of the GO graph and its organism-specific gene annotations. An overview on the structure of the XML file is given using as an example the organism-specific annotations of *Escherichia coli* (strain K12). According to the XML syntax, starting tags $< ... >$ and closing tags $< /... >$ specify the elements, e.g. the root element *graph*. Attributes are represented by green and the assigned values by red text color (e.g. $go\_no = "1"$). Three dots (...) symbolize place holders for XML code that had to be omitted due to space limitations since the original file contains more than 590,000 rows and has a size of about 16 MB.

class Prodoric_GenomReview_Uniprot_Adjustment triggered the assignment between the proteins from PRODORIC and those from UniProtKB. Here, an approach based on a pairwise sequence alignment was chosen. The Genome Reviews were taken as interlayer since they contain an up-to-date annotation of prokaryotic genomes, while maintaining cross-references to the primary sequence repositories such as GenBank, the source of submission of the genome sequences, and well-annotated data resources such as UniProtKB (Kersey *et al.*, 2005; Sterk *et al.*, 2006). In brief, for each organism the protein sequences were extracted from PRODORIC and the Genome Reviews. They were stored as FASTA files that were taken for a subsequent BLAST analysis. The BLAST database was generated from the Genome Review file and the PRODORIC protein sequences were used as queries (see chapter 2.4.4). The blastall program was run with the option *-m 8* to give a tabular output, in which multiple hits for the same query sequence were ordered ascendingly by their E-values. Subsequently, the best BLAST hits were extracted from the result file in a stringent approach (threshold: E-value $< 10^{-8}$, identity $> 60\%$) to identify for all PRODORIC proteins the corresponding proteins from the Genome Review. Exploiting the established PRODORIC-to-Genome Review protein assignment, the first UniProt accession numbers were extracted from the Genome Review. They were taken for the identification of the respective protein entries from UniProtKB yielding the PRODORIC-to-UniProtKB protein assignment. Finally, for each protein its official name and OLN were extracted from the in-memory UniProtKB representation, followed by the short name and OLN from in PRODORIC (columns 3 and 4) and by alternative gene names and synonyms from UniProtKB. The resulting file contains the compilation of gene names, OLNs, alternative names and synonyms for each protein of the selected organism in a tab-delimited form.

The second step, the matching of the gene names from the microarrays to the polypeptide assigned to the GO graph, occurs during an online JProGO analysis. The file with the compilation of gene names and synonyms of the analyzed organism whose data should be analyzed is read in. In order to avoid ambiguities, redundant synonyms, which occur in more than one row of the file are excluded from matching (class MicroarrayGenSynonymFilter). Then, for the recognition of the appropriate polypeptide names assigned to the GO graph the following priority order is maintained: The gene names from the microarray datasets are first matched with the official short names derived from UniProtKB (first column). For the genes which do not match in this step, the OLNs are considered next (see Scheer *et al.*, 2006). Finally, if necessary, the remaining alternative gene names and non-redundant synonyms are taken into account.

2.7.4 Statistical Analysis and Algorithms

The statistical analysis of the GO nodes with respect to their gene expression profile (chapter 1.4.3) is an important step in the functional interpretation of expression data with JProGO. In the following, first the basic required statistical terms, null hypothesis and alternative hypothesis, are introduced. Then the technical aspects for the statistical analysis of the GO nodes are outlined. This comprises, firstly, the conduction of the statistical testing itself and secondly, the correction for the multiple testing effect (see chapters 1.4.2 and 1.4.3 for background information).

Statistical background

The alternative hypothesis and the null hypothesis are the two complementary hypothe-

ses, whose probabilities are compared with a statistical test, e.g. the t-test. Normally, the alternative hypothesis represents the option that an observed effect is true, e.g. in the case of the t-test the means of two empirical normal-like distribution are different. Reciprocally, the null hypothesis denotes the other possibility that the effect has resulted just by chance and no true difference, e.g. in the mean or standard deviation of two empirical distributions, exists. Besides refusing the equality of a random samples' parameter such as the mean of two distributions, the alternative hypothesis can also restrict to only the deviation in one direction either the *greater than* or *smaller than*. Thus, the mean of one of the two distributions can either be greater than or smaller than that of the other. In the latter case, the alternative hypothesis is called one-sided, otherwise two-sided.

In statistical testing the probability is computed that the observed effect takes place given the null hypothesis is valid. If this value, also known as p-value, is small enough, that is below the selected level of significance, the null hypothesis is rejected in behalf of the alternative hypothesis.

Statistical testing and computation of the p-values

After identifying the official names and OLNs of the gene products encoded by the genes of the microarray (chapter 2.7.3), these were transfered to the expression matrix (2D array), to replace the synonyms and alternative gene names. This array of gene names and their expression levels is then used as input for the statistical testing of the nodes by an instance of the Analyzer class. The Analyzer object performs the following tasks (method run): First, a matching of the genes and their expression levels from the 2D array to the individual GO nodes is performed via their BitArray instances.

1) In case of a threshold-based test (Fisher's exact test), the number of genes that match the threshold criterion – with an expression level either above or below the chosen cut-off value and that are assigned to the GO node of interest – are determined. This number corresponds to the parameter k in the formula for Fisher's exact test (Eqn. 6), and the other parameters are also determined: the total number of measured genes (N), the total number of genes matching the threshold criterion (n) and the total number of (measured) genes that are assigned to the GO node (K). In the first version, Fisher's exact was implemented in *Java*. In more recent versions, the test was also performed using R and Rserve (see chapters 2.3.1, 2.3.2 and below for details).

2) In case of a threshold-free test, e.g. the Student's t-test, the two complementary distributions of gene expression values are determined: a) the distribution of expression values derived from genes assigned to the actual GO node and b) the background distribution derived from the remaining genes (see Fig. 5). They were stored in two arrays, in_node and out_node. From *Java* these two arrays were assigned to analogous data structures in *R*, which was run in a server mode using Rserve (chapter 2.3.2). The following commands were used in this context.

For the establishment of a connection between *Java* and *R* on the the local web server computer (chapter 2.1) using port 6313:

```
Rcon = new org.rosuda.JRclient.Rconnection("localhost", 6313);
```

For the assignment of the two arrays to the workspace of R:

```
Rcon.assign("x", in_node);
Rcon.assign("y", out_node);
```

For the computation of a GO node's p-value using the specified statistical test (variable stat_text, possible values: "ks.test", "wilcox.test", "t.test") and the chosen alternative hypothesis (member variable alternativeHypothesis: "less", "greater", "two.sided"):

```
org.rosuda.JRclient.RList Rlist;
...
String stat_test = ...
String this.alternativeHypothesis = ...
...
String Rcommand = stat_test + "(x,y,alternative=c(" +
                  this.alternativeHypothesis + ")";
Rlist = Rcon.eval(Rcommand).asList();
double p_value = Rlist.at("p.value").asDouble();
```

Statistical testing is restricted to GO nodes, for which a sufficient number of gene expression levels was measured, by default at least four genes. A smaller number was not regarded as meaningful and their theoretical p-values would almost never be significant.

Correction of the multiple testing effect
Since for each GO node a statistical test is performed related to the same set of expression data, a multiple testing problem arises (chapter 1.4.3) when computing the p-values. To cope with this effect, two methods of correction are offered in JProGO, the Bonferroni method and the FDR method (see chapter 1.4.3 for more information).
Bonferroni correction:
Assuming a constant nominal α error rate or p-value, which is valid for one comparison, the family wise error rate (FW) over N statistical tests is corrected exploiting the following relationships (see Bland and Altman, 1995): $\alpha_{FW} = 1 - (1 - \alpha_{nominal})^N$
and $\alpha_{FW} \leqslant N \cdot \alpha_{nominal}$
The corrected p-value is then simply computed as follows:

$$p_{cor} = \frac{p_{nominal}}{N}$$

FDR method:
A possible variant for the computation of the FDR is broadly outlined below (see also `http://www.unt.edu/benchmarks/archives/2002/april02/rss.htm`):

1. Choose α, the proportion of errors over the tests whose null hypothesis is rejected.
2. Create a vector A1 of the p-values, which have to be sorted.
3. Create a second vector A2 of the same length N of vector A2 which contains $j \cdot \frac{\alpha}{C_N \cdot N}$ whereas j=1,2,...,N and C_N is a constant ($C_N = 1$ for independent and $C_N = \sum_{i=1}^{n} \frac{1}{i}$ for dependent tests).
4. Subtract vector A1 from A2 and call the result vector B.
5. Search for the largest index d of the p-values whose value in vector B is negative (P_d).

6. Reject the null hypotheses of those tests having a p-value $\leq P_d$.

Benjamini and Hochberg (1995) showed that $FDR \leq \alpha$ for p_d where $d = max\{j : P_j \leq \frac{j \cdot \alpha}{C_n \cdot N}\}$ (see `http://www.unt.edu/benchmarks/archives/2002/april02/rss.htm`).

2.7.5 Visualization of the Results

The results of a JProGO analysis typically comprise several hundreds of GO nodes and their p-values. Two forms of representation were chosen for their interactive visualization: a) a tabular view and b) a subgraph representation (see chapter 3.1.2). Their implementation is described below.

Table view:
Two *Java* classes were created for the tabular representation of the results of an analysis: TableRow, which represents a single row of a table, and Table, which holds a java.util.Vector of TableRow instances. The Table class contains several sorting methods, sortDesc(int orderingColumn) and sortAsc(int orderingColumn), that allow to sort the table by a specified column such as p-value or GO node name. For this purpose, the sort method of the java.util.Collection class was used and the required interface Comparable (method compareTo()) was implemented by the TableRow class for the three different *Java* data types Integer (GO node no), String (GO node name) and Double (p-value). The search functionalities of the Table class comprise a fast seeking method which is based on a Hashtable. In the name field of each GO node a link to the extended node view site (see below) was inserted using an HTML form.

Subgraph view:
According to the corrected level of significance (chapter 2.7.4), for the subset of significant GO nodes all paths from them to the root node were computed. An algorithm based on the breadth-first search (BFS) is used in this context. The adapted algorithm comprises the following steps:

1. Put the selected node (class Node) in a first in, first out (FIFO) queue (here represented by java.util.Vector).

2. While the queue is not empty, remove the first node from the queue and determine whether it has already been visited before.

3. If the node has not been visited before, mark it accordingly as "visited" and inspect all its parent nodes (method getParents() from class Node) while memorizing the connecting edges.

4. Put the parent nodes that are not marked as "visited" to the end of the queue and repeat from step 2.

5. Finally, return the found subgraph, which comprises the visited nodes and edges

The BFS automatically terminate at the root node of the GO graph since for this node the method getParents() (step 3) returns none. In addition, the GO category of the selected node is determined by inspecting the names of all visited predecessor nodes which terminates at "molecular_function", "biological_process" or "cellular_component".

Since for each of the set of significant nodes a subgraph is returned, these graphs are merged and converted to the dot format that serves as input for the graphviz graph layout tool (chapter 2.4.5). Here, its name and a link to the corresponding extended node view site (see below) is used as node label. Furthermore, the node's size and brightness attribute is set inversely proportional to its p-value and the color attribute reflects the GO category. The subgraph is drawn with the graphviz tool using the generated dot input file and displayed as an HTML site or as a PDF file.

Extended node view:
The two above described representations of results, the Table and Subgraph view, give an overview on all analyzed GO nodes. In addition, both provide for each GO node a link to a more detailed view showing the genes assigned to it and the expression levels as well as the resulting gene expression profile (chapter 3.1.2). For the GO node's expression matrix (Fig. 16), instances of the classes Graph and Node are employed in order to obtain the assigned gene products. Since the polypeptide numbers of the PRODORIC database are used, the access to the genes of PRODORIC is straightforward. The accession numbers are retrieved and appended for each gene to the URL `http://www.prodoric.de/gene.php?gene_acc=` in order to provide a direct link to the gene view site of PRODORIC (see Fig. 10).

The two distributions of the expression values of the genes assigned to the GO node and the remaining ones (background distribution) are visualized as histograms of relative frequencies using the classes ExtendedGoView and JFreeChartHisto (see Fig. 17). The JFreeChartHisto class incorporates functionalities of the JFreeChart package (chapter 2.3.1) including the JFreeChart class itself and the class HistogramDataset to generate an appropriate XY-plot and png image.

2.7.6 Creation and Run of the Web-based Service

In the following, a short overview on the technologies and tools is given that were applied for the establishment and run of the JProGO web-based service. Since the main program was written in *Java*, the smooth integration of these classes was a main goal for the development of the web interface. Therefore, Java Servlet and Java Server Page (JSP) technology was utilized (chapter 2.3.1). The created Java servlet classes reflect the workflow of the analysis (Fig. 13). The start form was programmed in JSP which – using the FileUpload package (chapter 2.3.1) – uploads all data required for the analysis (see Fig. 12) and triggers the recognition of the gene names (chapter 2.7.3). It also redirects all data to a servlet which performs several checks such as controlling the validity of the number formats of the expression levels. If the user decides to proceed with the analysis, the expression data and other parameters are forwarded with the help of hidden HTML fields to the goanalyzerservlet class. This servlet triggers the actual statistical analysis of the GO nodes by communicating with the Analyzer class (chapter 2.7.4). The next step, the representation of the outcome of the analysis, is coordinated by the GOTableView servlet which integrates the functionalities of the classes responsible for the tabular and subgraph view (chapter 2.7.5).

After describing the creation of the JProGO web-based service and the web interface in particular, the preconditions for running the service are outlined below. The actual web server (state of December 2007, hardware see chapter 2.1) comprises the following software components:

- Operating system Ubuntu Linux 7.04, kernel version 2.6.20-16 (see chapter 2.2)
- *Java* HotSpot™server virtual machine (build 1.5.0_11-b03, mixed mode)
- Graphviz/Dot version 2.8
- *R*, version 2.3.1
- Rserve, version 0.3-17
- Jakarta Tomcat web server, version 5.5.9

In addition, to the *Java web archive* (war), which contains all JProGO classes, the GO graph XML files for the supported species (see Tab. 2.7.3) and the gene name-to-synonym assignment files (chapter 2.7.3) were deposited in the webapps directory of the Tomcat server.

2.8 Preprocessing of Microarray Gene Expression Data with Bioconductor

The preprocessing of all microarray raw expression data, which solely comprised Affymetrix GeneChip® expression data from *P. aeruginosa* (see chapter 2.5.1 for details), was done with Bioconductor. It is a free software framework based on and designed for the programming language *R* (for version information see chapter 2.3.2). The following Bioconductor libraries were used routinely for this purpose: affy, simpleaffy and vsn (chapter 2.3.2). The Affymetrix raw data files (CEL, CHP and DAT) were put into one directory together with an assignment file, in which the array file names were related with the experiment description (description.txt). These data were then loaded into the workspace of *R* including all replicate measurements per condition (e.g. 3) with the help of the read.affy() command (simpleaffy package):

```
raw.data <- read.affy(covdesc="description.txt",
path="<path to the CEL file directory>")
```

After a visual inspection of the raw images, which consist of the log-transformed intensities, the preprocessing was conducted. The applied preprocessing algorithms – in the sense of combinations of a specific background correction, normalization and summarization method (see chapter 1.4.1) – comprised rma, vsn and dChip (see also chapter 3.3.1). For this purpose, the expresso function of the affy package was employed with the parameters specified below.
1) rma method:

```
eset.rma <- expresso (raw.data,
bgcorrect.method = "rma",
normalize.method = "quantiles",
pmcorrect.method = "pmonly",
summary.method = "medianpolish")
```

2) vsn method:

```
eset.vsn <- expresso (raw.data,
bg.correct = FALSE,
normalize.method = "vsn",
```

```
normalize.param = list (subsample =10000),
pmcorrect.method = "pmonly",
summary.method = "medianpolish")
```

3) dChip method:

```
eset.liwong <- expresso (raw.data,
normalize.method = "invariantset",
bg.correct = FALSE,
pmcorrect.method = "pmonly",
summary.method = "liwong")
```

The preprocessed data sets were stored as text files in the CSV format, which were converted afterwards to spread sheet software files (Open Office 2.0).

Several diagnostic plots were created for the preprocessed and, for comparison, for the raw data using log-transformed values: RNA degradation plots (see Fig. 30) were computed to check whether the different measurements show deviations in their degradation profile. Box (see Fig. 24) and density plots (see Fig. 31) were generated for comparing the intensity ranges of the arrays and to control the success of normalization and comparability between the arrays. In order to assess the conformance and reproducibility of the replicate measurements, scatter plots of the different pairwise combinations were created (see Fig. 32). In these plots, in addition to the diagonal line (same expression levels of both measurements), parallel lines shifted by one and two log steps were included to visualize major deviations (log ratios) in the expression levels of the same genes. Histograms were generated for the scatter plots showing the frequencies of the log ratios. In addition, all pairwise Pearson's correlation coefficients were computed as a further quality criterion for the agreement between the replicate arrays. Heatmaps were created (heatmap command) as a result of hierarchically clustering the genes and, at the same time, the conditions. The latter clustering allowed to assess the similarity of the experimental conditions (replicate measurements) and to identify different conditions that show a similar expression profile. In order to separate random from non-random effects in the heatmap analysis, in each case, in addition to all probe sets present on the Affymetrix chip, the following two subsets were utilized: 14 negative control genes – from other organisms than *P. aeruginosa* – and the same number of randomly chosen genes.

2.9 Mid-level Analysis of Microarray Expression Data Using CyberT

The *R* package bayesreg (see chapter 2.3.2 for version information) was used for the determination of the ppde (posterior probability of differential expression). They were computed according to the CyberT algorithm (chapter 1.4.2). Only genes were considered whose preprocessed expression levels were available in all meaningful pairwise comparisons of conditions. As recommended in the help file of this package (bayesreg.readme), all data were transformed to logarithms prior to the computation. The regularized t-test was invoked by the bayesT command using the recommended betaFit value of 1, the Bayesian version of the test, a window size of 101 (number of neighboring genes) for obtaining the background variance and a weight of 10 giving to the Bayesian prior estimate. The latter parameter was chosen about 3 times the number of replicates, since with such a ratio the authors of the bayesreg package had observed a good performance (see readme file).

After the computation of the ppde, a volcano plot (see chapter 3.3.2 and Fig. 26) was generated to visualize the correlation of the pde and the expression ratios at once. For this purpose, the log_2-transformed expression ratios (X-axis) were plotted against the log_{10}-transformed ppde (Y-axis) using the plot command of *R*. In addition, colored circles were drawn around the 25 genes with the lowest and the 25 with the highest log ratio and around those with the 50 best ppde (using the point command). In addition, a legend of gene names for the easy identification of these genes was created (see Fig. 26).

2.10 Functional Interpretation of Microarray Expression Data with JProGO

The workflow (Fig. 13) and application of a JProGO analysis is described elsewhere in this thesis (chapters 3.1.2.5 and 3.3.3). Therefore, in the following practical aspects that are representative for typically performed analyses are summarized.

Before uploading the data, it was made sure that either OLNs or gene short names were included in the preprocessed expression matrix file whereas OLNs were preferred since they are normally unambiguous. If necessary, corresponding substrings containing only the OLNs were generated from the probe set names like for Affymetrix IDs of the GeneChip® array of *P. aeruginosa* (PAO1): Here, a substring containing the first six letters was generated (e.g. by standard spread sheet software) in order to extract the OLNs. For example, the IDs *PA5419_soxG_at*, *PA5420_purU2_at* and *PA5421_fdhA_at* were converted to *PA5419*, *PA5420* and *PA5421*. In the next step, the expression data were uploaded using the input form of the web interface (Fig. 12), whereas the choice of the adequate organism and the appropriate data type (ppde versus expression ratios) were crucial points to guarantee a correct analysis. For the remaining parameters, if not stated otherwise like when different statistical tests should be compared with respect to the outcome of the analysis (chapter 3.2.2), the preset default values of the JProGO web interface were taken. These comprise the U-test as the statistical test, a two-sided alternative hypothesis, the FDR method for correcting the multiple testing effect and an α of 0.05 as level of significance (see Fig. 12).

After performing the data check and the analysis itself the results were inspected using always both, the tabular and subgraph view. For the subgraph view the default setting includes all three GO categories *Molecular Function*, *Biological Process* and *Cellular Component*. This setting was kept unless the number of GO nodes in the subgraph was higher than about 40. In this case, only one or two of the above mentioned GO categories were selected for displaying. The results of an analysis were saved in a text file in the CSV format (tabular view) and as a PDF file (subgraph view).

2.11 Expansion of JProGO towards JRegA

For the expansion of JProGO towards JRegA, which includes regulons and operons as biological groupings, on the one hand the *Java* classes of the JProGO framework were directly used or extended (e.g. Analyzer class). On the other hand, several new classes were developed which mainly allow an object-oriented representation of the corresponding biological entities stored in the PRODORIC database: the experimentally validated operons and regulons. Amongst others, a Gene, Replicon and Genome class were created as well as an abstract container class BiologicalGeneGrouping for managing various

groupings of Gene instances such as Operon and Regulon (both classes extend BiologicalGeneGrouping). With the method readFromProdoricDB() of the class Replicon, which is itself invoked by the identically named method of the Genome class, the PRODORIC database was queried for genes, operons and regulons of the organism of interest. Appropriate SQL statements were generated by the *Java* program and, as with JProGO, transmitted to the PostgreSQL database using the JDBC technology (chapter 2.3.1) The following statement, for example, was employed in order to obtain the operons of a replicon of interest:

```
SELECT DISTINCT transcr_unit.transcr_unit_acc,
transcr_unit.transcr_unit_name, gene.gene_acc FROM
transcr_unit, gene2transcr_unit, gene, replicon
WHERE transcr_unit.transcr_unit_no =
gene2transcr_unit.transcr_unit_no
AND gene2transcr_unit.gene_no = gene.gene_no
AND gene.replicon_no = replicon.replicon_no
AND replicon.acc_no = ...
```

The obtained object-oriented in-memory representations of the regulons – expanded by the genes that are member of the same operons as the regulated ones – were then taken for the subsequent analysis of the expression data using the Analyzer class (adapted from the JProGO project, chapter 2.7.4). Access to the threshold-free statistical algorithms Student's t-test, U-test and KS-test was performed analogously to JProGO with the help of *R* and Rserve.

A command-line prototype of JRegA was completed in order to evaluate the performance of the tool with expression data sets of transcription factor knock out strains for which a clear expectation on the expression profile was available (chapter 3.4.2). More complex and time-consuming adaptations necessary for the creation of a web-based version of JRegA are planned for the future and so far only a beta test version exists (chapter 3.4.2).

3 Results and Discussion

When starting with the work for this thesis no freely accessible tool was available that allows the straightforward Gene Ontology-based functional analysis of prokaryotic high-throughput expression data and that also integrates several alternative statistical algorithms. Since the number of gene expression profiling experiments on bacteria and archaea has rapidly grown during the last years, a special need for such a program has arisen in the microbial research community. For this reason, a novel software was implemented that meets the described requirements and that is freely accessible via the web (see below).

3.1 JProGO: A Software Suite for the Functional Context-based Analysis of Prokaryotic Gene Expression Data Using the Gene Ontology

Within the scope of this thesis the web-based software tool JProGO was developed for the functional interpretation of prokaryotic microarray gene expression data (Scheer *et al.*, 2006, `http://www.jprogo.de`) using the Gene Ontology (GO) as the functional classification system. It sets transcriptional alterations in relation to biological processes and functions by identifying the respective GO terms for the genes that are changed in their expression profile under the two conditions compared. The tool provides the possibility to use either some threshold-based or several threshold-free algorithms for the detection of relevant GO terms. An appropriate method for the correction of the multiple testing effect and a flexible visualization of the obtained results is offered. A direct applicability to expression data from prokaryotic organisms is facilitated by using the PRODORIC database as data basis for the genomes. The necessary expansion of PRODORIC, which amongst others comprised the integration of GO, was a prerequisite for the development of JProGO and is described in chapter 3.1.1. The program itself and its features are described in detail in chapter 3.1.2.

3.1.1 Integration of the Gene Ontology into the PRODORIC database as Data Basis for JProGO

The incorporation of the functional classification system GO and the assignment of the genes to the GO terms (GOA project) required the structural expansion of the PRODORIC database. This primarily included two novel relations for storing the nodes and the edges of GO, which represents a directed acyclic graph (see chapter 1.3.2), and another relation that allows to memorize the assignment between GO nodes and genes. Details on these structural changes are given in the *Materials and Methods* section (see chapter 2.6.1). The second step comprised the actual integration of the GO and GOA data into PRODORIC. For this purpose, the corresponding flat files were parsed with a self-developed shell script and inserted into the newly created database relations (see chapter 2.6.1).

In the next step, the sites of the PRODORIC web interface were adapted (mainly work of R. Münch, Technische Universität Braunschweig) in order to display the GO terms assigned to a selected gene (see Fig. 10). Furthermore, umbrella terms, also called GO classes, were introduced to expand the search options for genes (chapter 2.6.2). These

General Information	
PRODORIC Acc No.	GE00104771
ORF ID	b2827
Short Name	thyA
Gene Name	thymidylate synthetase
Protein Name (Acc)	ThyA (PR00104771)
Organism	Escherichia coli (strain K12)
NCBI Taxonomy ID	83333 (Taxonomy Browser)
Gene Ontology	GO:0003723 (RNA binding) GO:0016740 (transferase activity) GO:0004799 (thymidylate synthase activity) GO:0008168 (methyltransferase activity) GO:0006231 (dTMP biosynthesis) GO:0006417 (regulation of protein biosynthesis) GO:0009165 (nucleotide biosynthesis) GO:0006445 (regulation of translation)
Description	b2827

Figure 10: PRODORIC web interface offering functional information from the Gene Ontology

terms comprise all *molecular functions* at the third level and all *biological process* terms at the fourth level below the root node. The PRODORIC web form that displays the GO terms is shown in Figure 11.

3.1.2 Use and Features of JProGO

After describing the extension of the PRODORIC database, which was the obligatory preparatory work for the creation of JProGO (see previous chapter), the capabilities of the JProGO program are specified in the following. An overview on the most important customizable parameters of an analysis, representing at the same time the key features, is given by the input form of JProGO's web interface (Fig. 12). This site serves as the starting point for each analysis. The entire web front-end was created to allow for an immediate access to the JProGO core program without the need for installation (chapter 2.7). It was implemented with the *Java* Servlet technology, whereas the core program of JProGO was written in *Java* using *R* for statistical testing (chapters 2.3.2 and 2.7). The web interface allows the user to customize all desired settings and options for the analysis in the run-up. The parameters of the input form that represent key features of JProGO are outlined in detail below. The preprocessed microarray gene expression data of interest have to be uploaded in the CSV format (field Microarray File, Fig. 12). Then, the analysis can be customized according the user's demands by specifying the corresponding parameters: the used statistical test, the level of significance, the method of correction for multiple testing and the type of expression data.

3.1.2.1 Statistical Methods for the Detection of the Relevant GO Nodes

JProGO provides four different statistical methods for the identification of groups of genes with a significantly altered expression profile. Three of them are commonly used for the analysis of eukaryotic expression data. They comprise the threshold-based Fisher's exact test and the threshold-free KS- and t-test (see chapter 1.4.3). Moreover, as an

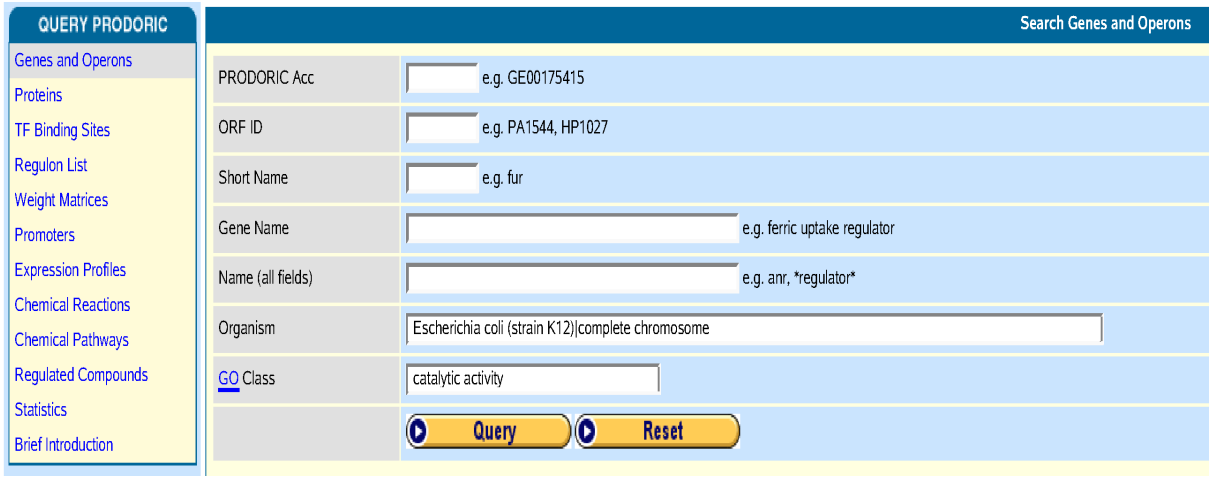


Figure 11: Expanded PRODORIC gene search form which allows to select upper level GO classes as search criteria. In this example screenshot, all genes that encode proteins with a *catalytic activity* (GO class) of *Escherichia coli* (strain K12) are looked up. The web site is accessible under http://www.prodoric.de/gsearch.php.

additional threshold-free test the rank-based unpaired Wilcoxon's test (U-test) can be chosen for a JProGO-based interpretation of expression data (Fig. 12). The implemented methods were selected by focusing on threshold-free methods since these better meet the requirements for a functional interpretation of microarray based gene expression experiments from a systems biology perspective for several reasons (see Dopazo, 2006). First of all, they take the continuous nature of gene expression into account and avoid the binarization of genes into the two classes of differentially and not differentially expressed genes. Thus, they use the whole available information on the consistently measured genes and obviate the sometimes enormous loss of information observed for threshold-based methods (Dopazo, 2006). Another reason which is related to the above mentioned is that through the avoidance of a threshold value, whose definition is always somehow arbitrary and which influences the outcome of the analysis, no groups of functionally related genes are split. Functionally related genes often simultaneously fulfill their roles in the cell and, thus, often bear a coordinated expression (Eisen *et al.*, 1998; Wolfe *et al.*, 2005; Dopazo, 2006).

One of the three threshold-free methods included in JProGO is the Student's t-test, which represents a parametric test. By contrast, both other threshold-free tests, the KS- and the U-test, are non-parametric and thus do not expect a Gaussian distribution for the expression profiles of the investigated GO terms. The U-test is, in addition, rank-based which may be advantageous if outliers are present among the gene expression values. In the JProGO input form it is the statistical method preselected by default (Fig. 12). With regard to the alternative hypothesis (see chapter 1.4.3 and 2.7.4) all statistical tests are provided in the two-sided and in both one-sided versions.

3.1.2.2 Correction of the Multiple Testing Effect

Since for each GO node that contains a sufficient number of genes for the organism of interest (see chapter 2.7.4) a separate statistical test has to be performed, a multiple testing problem arises. In order to compensate for this effect, two correction proce-

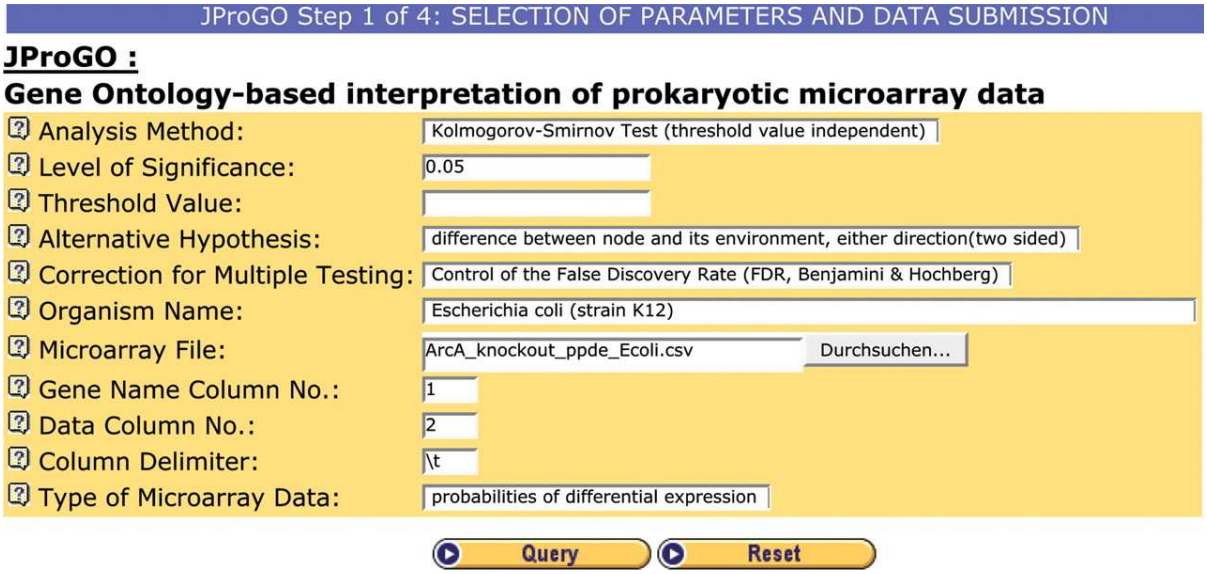


Figure 12: Screenshot of the input form of JProGO's web interface (taken from Scheer *et al.*, 2006)

dures are included. Either the control of the false discovery rate (FDR, see Benjamini and Hochberg, 1995; Benjamini and Yekutieli, 2001) or the Bonferroni correction are used (Bonferroni, 1936, see also Bland and Altman, 1995). The Bonferroni correction is conservative which bears the advantage that GO nodes found to be significant have a low probability of being false-positive hits. But the price to pay is that the number of significant hits often is underestimated, which can lead to the exclusion of many true positives. By contrast, the FDR method, which belongs to the most accepted methods of correction, has the advantage that its statistical power is greater than that of family-wise error rate (FWER) controlling methods like the Bonferroni correction (see Benjamini and Hochberg, 1995). Thus, the FDR method normally will identify additional GO nodes as significant which are missed by the Bonferroni method. It was chosen as the default method of correction for the multiple testing effect in JProGO (Fig. 12).

3.1.2.3 Supported Organisms and Matching of Alternative Gene Names

The tools available for the functional interpretation of microarray data mainly focus on the analysis of gene expression data from eukaryotes. Thus, they often cover important eukaryotic model organisms, such as man, mouse, rat, thale cress and yeast (see e.g. Boorsma *et al.*, 2005; Martin *et al.*, 2004; Zhang *et al.*, 2004). Only a few tools allow to customize their list of supported organisms for which purpose the user has to download and include the appropriate gene annotation files (e.g. Subramanian *et al.*, 2005). By contrast, JProGO supports the user-friendly immediate functional analysis of gene expression data from prokaryotes without any additional activities required from the user such as generating the assignment between the GO nodes and the corresponding genes. Currently, expression data from more than 20 prokaryotic species are supported (see Scheer *et al.*, 2006 and Tab. 5 in the *Material and Methods* section). The model bacterial organisms *B. subtilis* (strain 168) and *E. coli* (strain K12) as well as *C. glutamicum* (strain DSM 20300), *H. pylori* (ATCC 700392), *L. monocytogenes* (strain EGD-e), *M. jannaschii* (strain JAL-1), *P. aeruginosa* (strain PAO1) and *P. putida* (strain KT2440),

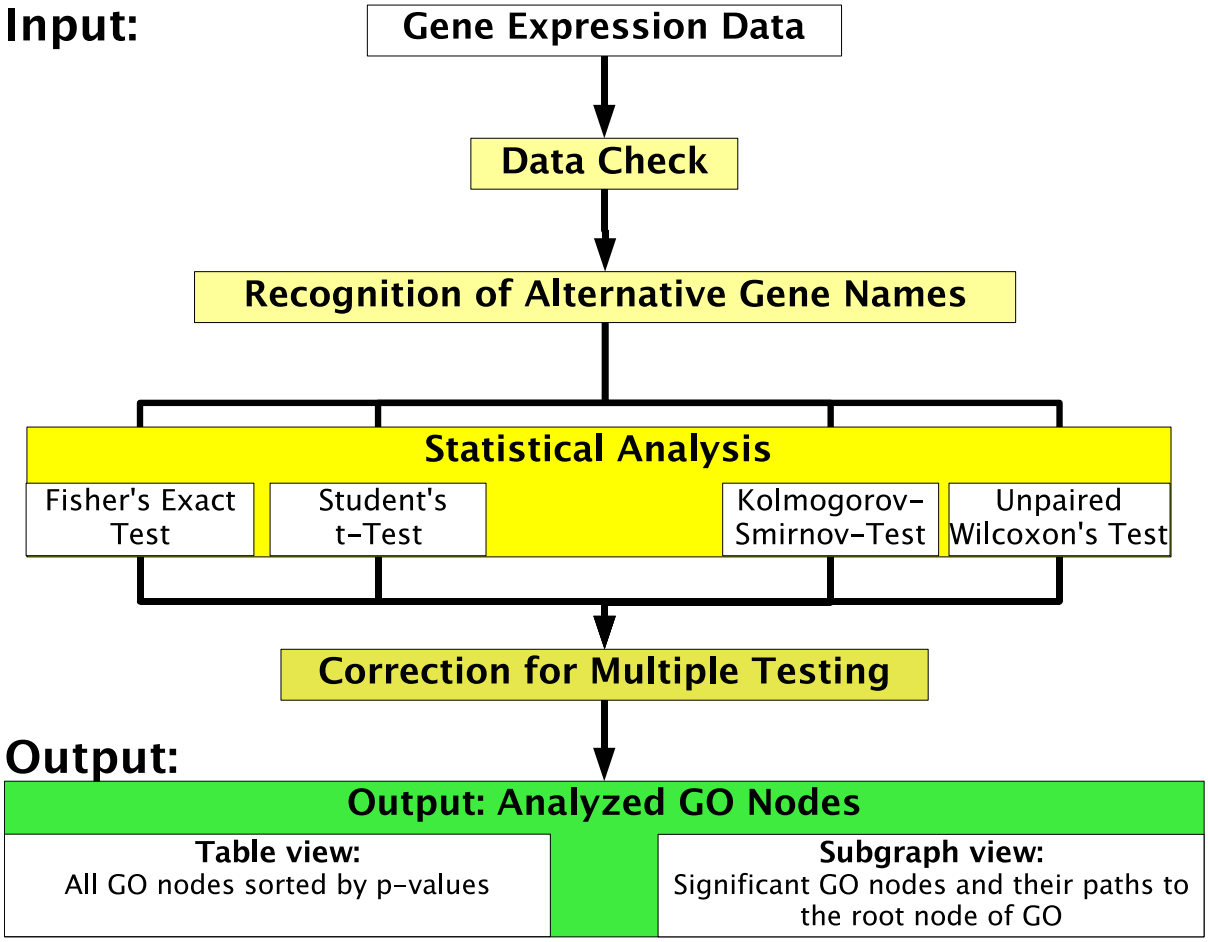


Figure 13: Sequence of analysis steps of JProGO with input and output data (adapted from Scheer *et al.*, 2006)

two *Rickettsia* strains, several pathogenic *Mycobacteria*, *Mycoplasma*, *Yersinia* and *Streptococcus* strains are included.

Another feature of JProGO is the automatic recognition of alternative gene names and synonyms for each supported organism. This is important because of the different nomenclatures that exist for naming genes and proteins (Koike and Takagi, 2004; Pillet *et al.*, 2005). As a consequence, the same entity is normally described by several different terms comprising, for example, the full name, the short name and synonyms. Thus, the inclusion of a matching strategy for gene names minimizes losses during the mapping of the gene expression data to the GO nodes. For this purpose, the following priority order is maintained. The gene identifiers from the microarray data sets are firstly matched with the official short names. Afterwards, the ORF IDs and synonyms are considered for the remaining identifiers (Scheer *et al.*, 2006). In order to avoid ambiguities, redundant synonyms are excluded. Gene short names, synonyms and ORF IDs were collected from PRODORIC (Münch *et al.*, 2003, 2005) and from the Uniprot database (Wu *et al.*, 2006) in the run-up (see chapter 2.7.3).

3.1.2.4 Accepted Input Data

JProGO accepts as input data both expression ratios and probabilities of differential expression (pde). While expression ratios have the advantage that they indicate the

direction of a change in gene expression – for example whether a gene is two-fold up- or down-regulated – they have the disadvantage that they do not consider the variance of the corresponding expression levels. The latter is taken into account by the pde, which therefore are recommended if a sufficient number of replicate measurements is available. The drawback of the pde is that they do not indicate whether a gene is up- or down-regulated.

3.1.2.5 Performing an Analysis and Visualization of the Obtained Results

For performing a functional analysis with JProGO, the preprocessed gene expression data have to be uploaded and the parameters described above should be specified (see Fig. 12 and 13). Then, the format of the expression values and the validity of the whole expression matrix is checked and its gene identifiers are matched with the genes of the selected prokaryotic organism (Fig. 13). Depending on the outcome of this process, the user may decide to proceed with the next step of analysis, the statistical evaluation of the GO nodes, or return to the data submission form and customize the parameters. In the first case, a p-value is computed for each considered GO node and afterwards the correction for the multiple testing effect is performed. This process typically takes a few minutes depending on the actual workload of the web server.

The results of the analysis are represented as an interactive table with several sorting and filtering functionalities (table view, Fig. 14). This table contains all tested GO nodes which are, by default, sorted by their p-values. Thus, the statistically significant nodes, if there are some, are shown at the top. The table view offers additional searching and sorting options. These comprise searching for GO nodes by their names, a p-value filter criterion and the restriction of GO nodes to one or two of the three sub-ontologies *Molecular Function* (MF), *Biological Process* (BP) and *Cellular Component* (CC, see also Fig. 4). Nodes meeting the corresponding criteria are high-lighted.

In addition to the table view, a second form of visualizing the results was implemented: the subgraph view (Fig. 15). This view consists of a subgraph which contains all significant GO terms as leave nodes and their paths up to the root node. The paths are computed with standard graph traversal algorithms based on a breadth-first search (see chapter 2.7.5). Furthermore, the size and brightness of each GO node is inversely proportional to its p-value, so that the relevant affected functions and processes are easily recognized. The visualization of the subgraph's nodes and edges is performed with the help of the graphviz package (see chapters 2.4.5 and 2.7.5 for details). The subgraph visualization is an important feature of JProGO, because the relevance of the significant GO nodes is shown in the context of their parent nodes, which represent more general functions and processes. A fast overview on the interrelationships of the results can be obtained this way.

Both, table view and subgraph view, provide direct access to the genes assigned to the individual GO nodes. Clicking on a node generates a new web page that contains a table with all genes assigned to this node and their expression levels (Fig. 16). In addition, the expression profile of the genes belonging to the GO node and those of the background distribution are visualized as histograms (Fig. 17). Furthermore, each gene provides a link to the PRODORIC database (Münch *et al.*, 2003, 2005), in which in-depth regulatory and functional information is offered. The results of an analysis can be downloaded as tab-delimited text file (table view) and as a PDF or PNG image (subgraph view).

Search and filter options

The following GO categories (sub-ontologies) are displayed:

Biological Process ☑ Cellular Component ☑
Molecular Function ☑

Input *level of significance* (p-value) or search text: 6.5241E-5

Selected Column: p-value

Compare Option: <=

Show Only Marked: no

Search / Filter

Reset

Number of nodes meeting the filter/search criterion: 10

Download as Text File

GO Category	GO Accession	GO Name	p-value
MF	GO:0005215	⊞ transporter activity	3.5544E-6
BP	GO:0006810	⊞ transport	7.1283E-6
BP	GO:0051179	⊞ localization	8.7035E-6
BP	GO:0051234	⊞ establishment of localization	8.7035E-6
BP	GO:0005996	⊞ monosaccharide metabolism	1.4171E-5
MF	GO:0003676	⊞ nucleic acid binding	1.7568E-5
BP	GO:0019318	⊞ hexose metabolism	2.415E-5
BP	GO:0006811	⊞ ion transport	3.8184E-5
BP	GO:0006351	⊞ transcription, DNA-dependent	3.9855E-5
BP	GO:0006066	⊞ alcohol metabolism	6.5241E-5

Figure 14: Table view of the output of a typical JProGO analysis. Only the significant GO nodes are shown together with their p-values (also sorted by p-values). As expression data the ppde from an *arcA* knockout data set of *E. coli* (Salmon *et al.*, 2005) were taken.

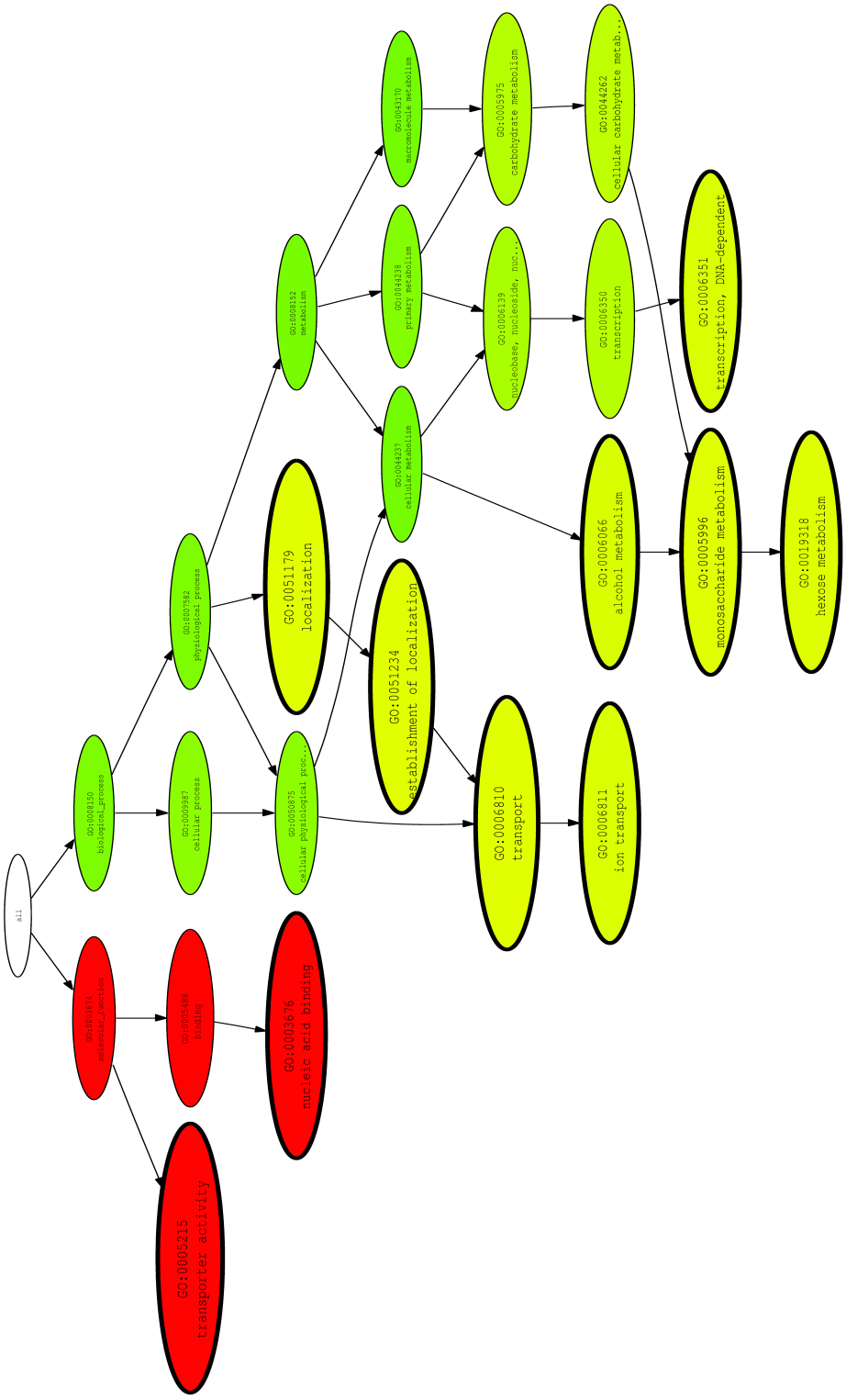


Figure 15: Subgraph view of the results of a typical JProGO analysis (rotated by 90°). Significant GO nodes are marked with a thicker border. They comprise the leave nodes of the subgraph and all paths up to the root node are computed. The larger and brighter a node is, the lower (better) is its p-value. The used data set and parameters of analysis are the same as in Fig. 14.

JProGO: EXTENDED GO NODE VIEW

Name of Gene Ontology Node:	hexose metabolism
Analysis Result:	2.415E-5

Go Directly To Expression Value Distributions

Assigned Genes:

Click on short names in order to get more information from PRODORIC database
Genes' expression values (microarray data) are shown in [brackets] where measured and assignable.

1) AceE [0.994983]	2) AceF [0.961621]	3) AceK [0.996790]	4) AgaY	5) B1773 [0.066282]	6) B2016 [0.000195]
7) B2097 [0.938910]	8) B2736 [0.147269]	9) Eno [0.992364]	10) Epd [0.471217]	11) Fba [0.999428]	12) FucA
13) FucI	14) FucK	15) FucO	16) FucP [0.999646]	17) FucR	18) FucU
19) GalE [0.213225]	20) GalF	21) GalK	22) GalM	23) GalR	24) GalT
25) GalU	26) GapA [0.999974]	27) GapC_1 [0.985204]	28) GapC_2	29) GatY [0.999845]	30) Glk
31) Gnd [0.900808]	32) GpmA [0.999909]	33) GpmB [0.999937]	34) IlvD [0.999976]	35) LpdA [0.999938]	36) MipB
37) PckA [0.874699]	38) PfkA [0.986982]	39) PfkB [0.991865]	40) PflA [0.000000]	41) PflB [0.978501]	42) PflC
43) PflD [0.999933]	44) Pgi [0.974760]	45) Pgk [0.999320]	46) Pgm [0.979236]	47) PpsA	48) PykA [0.049697]
49) PykF [0.982861]	50) RhaA [0.999919]	51) RhaB [0.999923]	52) RhaD	53) RhaR	54) RhaS [0.999379]
55) RpiA [0.935578]	56) RpiB	57) SdaA [0.996867]	58) SdaB [0.982529]	59) SucA [0.990214]	60) TalA [0.998077]
61) TalB [0.646664]	62) TalC	63) TdcE [0.961374]	64) ThiG [0.979992]	65) TpiA [0.648587]	66) XylA [0.999976]
67) YbbQ	68) YbgG [0.999972]	69) YbhE	70) YbiW [0.987871]	71) YbiY	72) YeaD [0.993951]
73) YhaE [0.99902]	74) YhaP	75) YhaQ	76) YhiE [0.997805]	77) YhjN	78) YhjO
79) YibO [0.000013]	80) YihR [0.998840]	81) YihS [0.998808]	82) YihU	83) YjjW	84) YphB [0.997610]

Figure 16: Extended view of a GO node. All genes assigned to the specified GO node are shown together with their expression values. In this example, the selected node is *hexose metabolism* to which 84 genes are assigned. The used data set and parameters of analysis are the same as in Fig. 14.

3.1.2.6 Distinction of the JProGO Approach from Related Tools and Methods

Most of the tools that were developed for the functional interpretation of microarray gene expression data (e.g. Boorsma *et al.*, 2005; Al-Shahrour *et al.*, 2004; Boyle *et al.*, 2004; Zhang *et al.*, 2004) offer only a single statistical test for the determination of the significant GO nodes that is either threshold-based or threshold-free (chapter 1.4.3). While several of these tools employ a statistical test that was applied earlier for a GO-based high-level analysis of expression data (e.g. Boyle *et al.*, 2004; Martin *et al.*, 2004; Zhang *et al.*, 2004), other tools propose a test procedure that was not used before in this context (e.g. Breslin *et al.*, 2004; Boorsma *et al.*, 2005; Smid and Dorssers, 2004; Subramanian *et al.*, 2005; Volinia *et al.*, 2004). By contrast, JProGO provides an integrative platform containing one threshold-based and several threshold-free statistical tests for the determination of the relevant GO nodes. This allows to perform an unbiased comparative analysis using more than just one or two statistical methods (see chapter 3.2). Moreover, the user of JProGO has the opportunity to choose or identify the best suited method for the interpretation of his/her data, e.g. to select a threshold-free non-parametric rather than a parametric test. With the help of JProGO the limitations of the threshold-based approaches were depicted in chapter 3.2.1. Furthermore, JProGO was used in a comparative evaluation of the three threshold-free statistical tests t-, KS- and U-test in the context of prokaryotic expression data and recommendations about which method to use were given (chapter 3.2.2).

Since most of the existing tools focus on eukaryotes, another distinctive feature of JProGO is its specific aptitude for the analysis of prokaryotic gene expression data (see

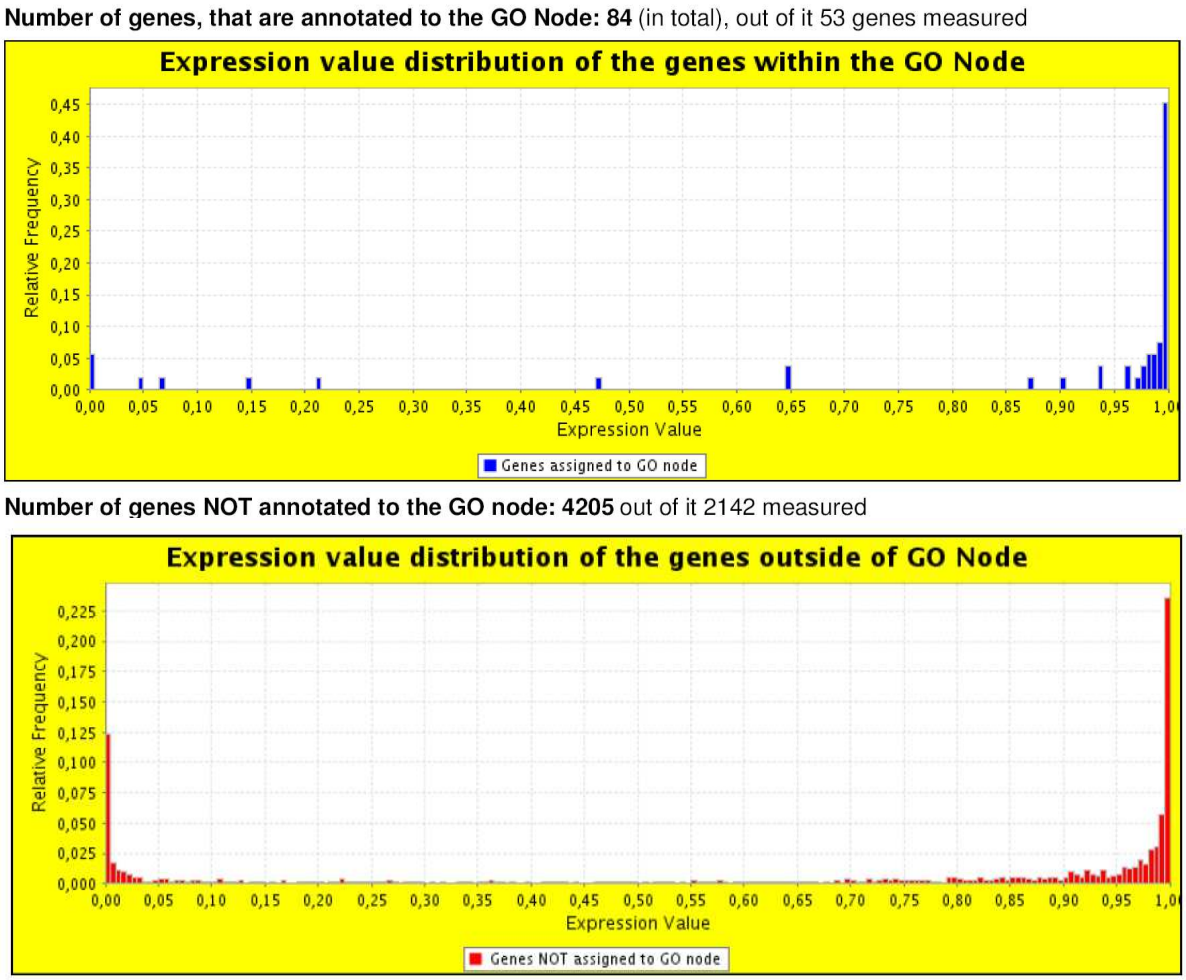


Figure 17: Extended view of a GO node. The distribution of expression values (ppde) of the genes assigned to the specified GO node and of the genes that are not assigned to the node are shown. In this example, the selected node is *hexose metabolism* to which 84 genes are assigned. Their ppde were derived from an *arcA* knockout data set (Salmon *et al.*, 2005)

chapter 1.4.3). This includes, as mentioned above, a comprehensive recognition of gene identifiers and synonymous names of prokaryotic genes as well as the pre-computation of the organism-specific annotations for the GO graphs of numerous bacterial and archaeal strains (Tab. 5). In this context, the linking of the genes and gene products of JProGO to the PRODORIC database, which provides several alternative names, the DNA sequences and further information, should be mentioned (see chapters 2.6 and 2.7.2). On the one hand this allows a straightforward expansion of the supported organisms and on the other hand the extension of JProGO towards other biological groupings such as operons and regulons (see chapter 3.4).

Since the publication of the JProGO manuscript, another application, FIVA (Functional Information Viewer and Analyzer), was developed that took up the idea of exclusively focusing on the evaluation of prokaryotic expression data (Blom *et al.*, 2007). Like JProGO, the FIVA tool offers a powerful recognition of prokaryotic gene identifiers, offers several methods for correcting the multiple testing effect, and aids, in contrast to JProGO, in the preprocessing of the raw expression data. However, it only provides one statistical test for the detection of the relevant nodes, which is the threshold-dependent Fisher's exact test. Besides the complete absence of any threshold-free statistical test, the FIVA tool neither allows for a subgraph-based representation of the obtained significant GO nodes.

3.2 High-level Analysis of Preprocessed Prokaryotic Gene Expression Data with JProGO

After the description of the JProGO program (see previous chapter), this chapter describes its application for a comprehensive analysis of published prokaryotic microarray gene expression data. One goal of this study was to evaluate both the similarities and differences between the individual statistical algorithms implemented in JProGO including potential advantages and drawbacks. In this context their general aptitude for the interpretation of prokaryotic expression data was investigated. In addition, the impact of the cut-off value on the results obtained with a threshold-based methods was elucidated. Furthermore, the influence of the type of input expression data on the outcome of a functional interpretation was investigated. For this purpose the two main types of expression data, test statistics (e.g. probabilities of differential expression) and expression ratios, were selected.

3.2.1 Limitations of Threshold-based Algorithms and the Impact of the Threshold Value

The first methods for the detection of interesting functional groups of genes, e.g. GO nodes, within microarray gene expression data were based on a threshold value that leads to a binarization of the genes into a list of differentially expressed genes – e.g. up- or down-regulated by more than two-fold – and the remaining genes that were considered as not differentially expressed (see Zeeberg *et al.*, 2003; Martin *et al.*, 2004; Zhang *et al.*, 2004 and chapter 1.4.3). These methods are still often used (see Khatri and Draghici, 2005), but only two studies are known that examined the influence of the threshold value using simulated data (Pan *et al.*, 2005) or eukaryotic expression data derived from tumor cells (Breslin *et al.*, 2004; Pan *et al.*, 2005). Therefore, the influence of the threshold

value on the analysis outcome of prokaryotic expression data was investigated in detail. In contrast to Pan *et al.* (2005), who used test statistics as input data, the common expression ratios, which can be computed for any number of replicates, were employed as data type. They were obtained from the publication of Kang *et al.* (2005). In this paper, the authors compared an *E. coli fnr* mutant with the respective wild type under aerobic and anaerobic conditions. Fnr is the oxygen regulatory protein of *E. coli*. Fisher's exact test (two-sided alternative hypothesis) was employed for the detection of the significant GO nodes. For this purpose, a significance level of 0.05 was used and the Bonferroni method was taken for the correction of the multiple testing effect. As a typical cut-off value to consider a gene as up-regulated an expression ratio of 2.0 was chosen (see Fig. 18). This cut-off is according to the two-fold rule introduced by Schena *et al.* (1995) and is commonly used in the literature, especially when only one or two replicate microarray measurements were performed. A differentiated discussion of this rule can be found in Hung *et al.* (2002) and Hatfield *et al.* (2003).

Biological processes that are generally affected upon adaptation to anaerobic growth conditions such as '*glycolysis*' and '*alcohol catabolism*' were found to be statistically significant GO nodes for this cut-off value. Several genes of these processes seem to be independent of the transcriptional regulator Fnr (see also Kang *et al.*, 2005). Then, the threshold value was varied from 1.7 to 2.3 in one-tenth increments and the significant GO nodes were determined using the same parameters of analysis as for the initial threshold value of 2.0. Figure 18 shows that the number of significant GO nodes does not remain constant: a decrease of the initial cut-off value (2.0) leads to a higher number and, vice versa, an increase of the cut-off value results in a lower number of significant nodes. However, lowering the threshold value does not necessarily cause a raise of the number of significant GO nodes (see e.g. cut-off 1.9 and 1.8 in Fig. 18), and increasing the threshold value does not always lead to a reduction of its number (see e.g. cut-off 2.2 and 2.3 in Fig. 18). This finding corresponds with the behavior deducible from the equation of Fisher's exact test (Eqn. 5) and with the results of Pan *et al.* (2005). Another observation is that most of the threshold values show a characteristic composition of significant GO nodes (Fig. 18). For example, when the initial cut-off is lowered only by one tenth to 1.9, the number of significant nodes increases from 12 to 18. Additional significant biological processes such as '*energy derivation by oxidation of organic compounds*' and '*oxidoreductase activity, acting on hydrogen as donor*' appear. These additional nodes fit the general expectation when comparing aerobic and anaerobic cultivation conditions and the expectation of the authors of the expression profiling study (compare to Kang *et al.*, 2005, too). Conversely, only a slight increase of the initial threshold value to 2.1 and 2.2 drastically reduces the number of significant nodes from 12 to 8 and just 2, respectively, erasing GO nodes that clearly fit the expectation (Fig. 18). Altogether, these findings exemplify the arbitrariness of the definition of a cut-off value for the selection of differentially expressed genes. This arbitrariness can lead to completely different biological interpretations of the identical expression data set even when the threshold value is only slightly changed. Of course, this does not only apply to expression ratios but also to test statistics such as the probabilities of differential expression, for which a cut-off probability must be defined (see Pan *et al.*, 2005). Thus, the essential drawback of the threshold-based methods is that they need a predefined and necessarily arbitrarily chosen cut-off value which strongly affects the outcome of the analysis. The results for the tested *E. coli* data are consistent with those of Pan *et al.* (2005), who illustrated a strong effect of the cut-off p-value for eukaryotic expression data. These authors also

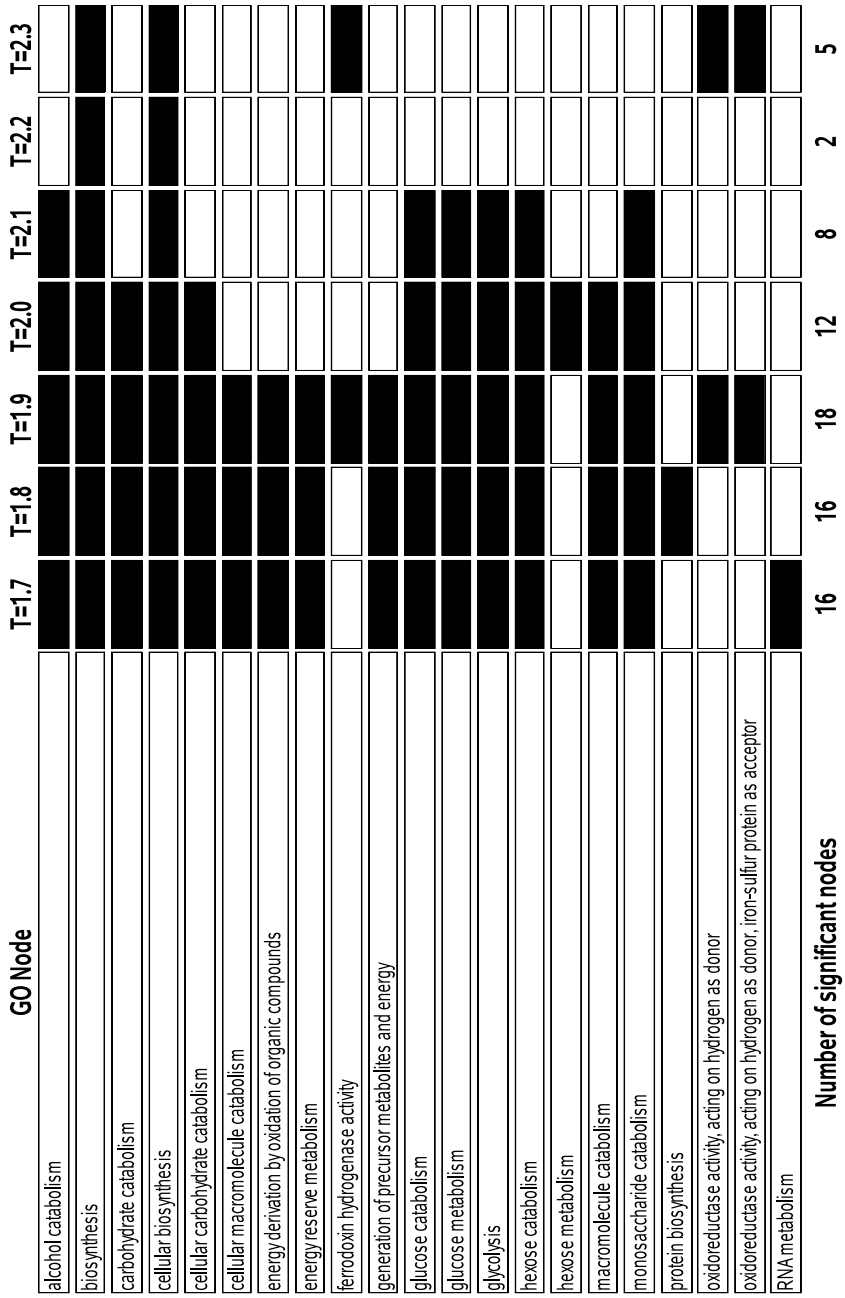


Figure 18: Arbitrariness of the cut-off value and its impact on the composition of significant nodes in the functional interpretation of prokaryotic gene expression data. A range of cut-off values (T) from 1.7 to 2.3 was used. As threshold-based test Fisher's exact test was chosen (two-sided alternative hypothesis) and the same expression data set (*fnr* knockout strain of *E. coli* from Kang *et al.*, 2005) was employed for all cut-off values. Significant nodes obtained with a certain threshold value are represented as black filled rectangles and their numbers are specified at the bottom.

observed a similar tendency that the effect of the chosen threshold value on the number of significant categories (GO nodes) is not linear (Pan *et al.*, 2005). Thus, increasing the threshold value might first lead to a reduction of the number of significant nodes, but a further increase could also raise this number again (see Fig. 18).

3.2.2 A Comparative Case Study Using Expression Data from *E. coli* K-12

3.2.2.1 Design of the Study and Selected Expression Data

Motivation and background:
The fundamental disadvantage of the threshold-based methods, that is the strong influence of the arbitrary cut-off value (see chapter 3.2.1 and Pan *et al.*, 2005), was the motivation for focusing on the evaluation of threshold-free methods. They were introduced more recently in the context of the functional interpretation of gene expression data and may constitute a more appropriate alternative. Most publications in this field either introduce a particular threshold-based or threshold-free method and attempt to show the general superiority of it, or re-implement a previously published method. By contrast, in the following a case study is described that investigates and directly compares the performance of three of these threshold-free algorithms on the same data sets (see below). The only known comparable evaluation of cut-off free methods was carried out by Breslin *et al.* (2004), who used a small number of three eukaryotic expression data sets (with 8 different data sets the current study takes more than twice as much). Breslin *et al.* (2004) tested the U-test, the GSEA method (Mootha *et al.*, 2003), which is similar to the KS-test, and the *minimal cutoff-based p-value* method (Berriz *et al.*, 2003; Breitling *et al.*, 2004). In addition to the rank-based non-parametric U-test, the original version of the KS-test, which was introduced by Ben-Shaul *et al.* (2005) as non-parametric test for the interpretation of expression data and was not checked against the U-test before, was included. Furthermore, the parametric t-test was chosen since its performance in the knowledge-based analysis of expression data was not compared to the other two non-parametric methods before. Among the different analyzed statistical methods (the GSEA variant of KS-test, no t-test), Breslin *et al.* (2004) employed separate tools for performing the individual tests, which makes a direct comparability of the results more complicated, whereas the case study at hand uses the integrative program suite JProGO for all the evaluated three cut-off free tests. Furthermore, in contrast to Breslin *et al.* (2004), the JProGO-based study only considers GO categories to which at least four measured genes were assigned and not just one or two. It also does not solely compare the ranks obtained for the GO nodes but their actual p-values. This allowed to compute both Spearman's rank and Pearson's correlation coefficients for the pairwise comparisons between the different methods (see Tab. 8).

Selected prokaryotic expression data sets:
E. coli (strain K12) constitutes the best annotated prokaryotic model organism with the highest fraction of genes to which at least one GO term was assigned (around 65% of all genes). In addition, a large number of expression data sets are available for this model organism (see e.g. GEO database, Barrett *et al.*, 2007). Therefore, *E. coli* was chosen to evaluate the threshold-free algorithms in a comparative case study using JProGO. The selected data sets should meet the following demands:

Table 6: Preprocessed expression data selected for a case study in *E. coli* in order to evaluate the threshold-free statistical tests implemented in JProGO

Investigated conditions	Data type	Reference	Recognized genes
arcA$^-$ vs. Wt (anaerobic growth)	ppde	Salmon *et al.*, 2005	2002 out of 2264
fnr$^-$ vs. Wt (anaerobic growth)	ppde	Salmon *et al.*, 2003	2338 out of 2402
fnr$^-$ vs. Wt (anaerobic growth, N_2)	p-values	Kang *et al.*, 2005	4167 out of 4337
fnr$^-$, aerob vs. *fnr*$^-$, anaerob (N_2)	p-values	Kang *et al.*, 2005	4167 out of 4337
fnr$^-$ vs. Wt (aerobic growth)	p-values	Kang *et al.*, 2005	4167 out of 4337
lrp$^-$ vs. Wt	ppde	Hung *et al.*, 2002	2690 out of 2758
Wt, aerob vs. Wt, anaerob	ppde	Salmon *et al.*, 2003	2748 out of 2820
Wt, aerob vs. Wt, anaerob (N_2)	p-values	Kang *et al.*, 2005	4167 out of 4337

- A sufficient number of replicates must be present for a reliable assessment of the probabilities of differential expression (pde).
- The data should be already preprocessed by the authors in a reliable and comparable way (e.g. either expression ratios or test statistics).
- The authors should formulate clear biological hypotheses about the expected outcome of their experiment.
- Major or global changes in expression should in all likelihood occur to obtain enough differentially expressed genes and hence a sufficient number of significant GO nodes for the comparison of the individual methods.

The first three criteria were met by the 8 preprocessed expression data sets listed in Table 6. The fourth condition was also expected to be fulfilled since the investigated conditions comprise either a knockout of a gene for a global transcriptional regulator (e.g. ArcA, Fnr or Lrp) or a strong alteration of the cultivation conditions such as the switch from aerobic to anaerobic growth (Salmon *et al.*, 2003, 2005; Kang *et al.*, 2005), or both. Deliberately, in two cases, which include the aerobic vs. anaerobic growth of wild type cells and the knockout of the *fnr* gene vs. wild type cells (see Tab. 6), two data sets from different authors with similar investigated conditions were included. Furthermore, with the experiment 'wild type cells vs. *fnr* knockouts, both cultivated aerobically' a kind of a negative control was enclosed. For this, besides the inevitable technical and biological variance, no broad perturbation of the gene expression profile is expected, because the regulator Fnr functions as O_2-sensor and is transcriptionally active only under anaerobic conditions (see Gunsalus and Park, 1994; Kiley and Beinert, 2003).

Parameters for the JProGO-based analysis:
For the case study the ppde and p-values, respectively, were extracted from the selected data sets (see Tab. 6) and used as input data for the analysis. A significance level of 0.05 was employed and the FDR method was taken to correct the multiple testing effect. Each expression data set was analyzed with a two-sided alternative hypothesis using the three threshold-free methods mentioned above (U-, KS-, t-test). The results of the analysis (see Tab. 7, 8, 20 and 21) are presented and discussed in the following chapters. In this context, firstly, a descriptive-statistical (chapter 3.2.2.2) and, secondly, a biological comparison and assessment of the methods was performed (chapter 3.2.2.3).

3.2.2.2 Statistical Evaluation and Comparison of Threshold-independent Methods

Starting with the inspection of the absolute numbers of GO nodes marked as significant under the chosen parameters ($\alpha = 0.05$), for all 8 data sets the t-test found the highest numbers of significant nodes (Tab. 7). The U-test followed at the second position with two exceptions, in which it identified the same or a slightly lower number than the KS-test, that otherwise detected the least numbers (Tab. 7). The observation that none of the three methods identified any significant node for the conditions 'fnr^- vs. Wt (aerobic growth)' may be explained by the fact that the data set constitutes a kind of negative control, for which only minor changes in the gene expression profile are expected (see chapter 3.2.2.1). How many of the nodes that were marked as significant for the other

Table 7: Total number of significant nodes for the Kolmogorov-Smirnov test, Student's t-test and Mann-Whitney-U test (abbreviated by #KS, #t, #U) and the pairwise ratios of common significant nodes in relation to the respective set union (Jaccard index, abbr. by JI). The data sets analyzed refer to Tab. 6 (same order). Rows with no significant nodes (null control) were omitted for the computation of the mean, median and standard deviation.

Conditions & Reference	#KS	#t	#U	JI(KS,t)	JI(KS,U)	JI(t,U)
arcA$^-$ vs. Wt (anaerob) Salmon et al., 2005	10	14	10	0.091	0.818	0.091
fnr$^-$ vs. Wt (anaerob) Salmon et al., 2003	7	26	17	0.222	0.412	0.229
fnr$^-$ vs. Wt (anaerob) Kang et al., 2005	32	83	41	0.373	0.698	0.447
fnr$^-$, aerob vs.*fnr*$^-$, anaerob Kang et al., 2005	33	57	32	0.552	0.806	0.534
fnr$^-$ vs. Wt (aerob) Kang et al., 2005 *null control*	(0)	(0)	(0)	–	–	–
lrp$^-$ vs. Wt Hung et al., 2002	11	22	19	0.500	0.579	0.818
Wt, aerob vs. Wt, anaerob Salmon et al., 2003	28	42	40	0.321	0.650	0.306
Wt, aerob vs. Wt, anaerob Kang et al., 2005	21	30	28	0.529	0.690	0.579
Mean	20.3	39.1	26.7	0.370	0.665	0.429
Median	21	30	28	0.373	0.690	0.447
Standard Deviation	11.0	23.9	11.9	0.172	0.139	0.243

Table 8: Pairwise Pearson's (r) and Spearman's rank (R) correlation coefficients of the p-values of all GO nodes computed using the Kolmogorov-Smirnov test, Student's t-test and Mann-Whitney-U test (abbreviated by KS, t, U). The data sets analyzed refer to Tab. 6 (same order).

Conditions & Reference	r(KS,t)	R(KS,t)	r(KS,U)	R(KS,U)	r(t,U)	R(t,U)
arcA$^-$ vs. Wt (anaerob) Salmon et al., 2005	0.687	0.694	0.843	0.857	0.674	0.693
fnr$^-$ vs. Wt (anaerob) Salmon et al., 2003	0.658	0.673	0.848	0.875	0.688	0.720
fnr$^-$ vs. Wt (anaerob) Kang et al., 2005	0.807	0.837	0.854	0.885	0.876	0.892
fnr$^-$, aerob vs.*fnr*$^-$, anaerob Kang et al., 2005	0.766	0.805	0.803	0.841	0.805	0.831
fnr$^-$ vs. Wt (aerob) Kang et al., 2005	0.800	0.798	0.809	0.811	0.939	0.939
lrp$^-$ vs. Wt Hung et al., 2002	0.840	0.860	0.833	0.856	0.969	0.968
Wt, aerob vs. Wt, anaerob Salmon et al., 2003	0.694	0.709	0.846	0.871	0.798	0.823
Wt, aerob vs. Wt, anaerob Kang et al., 2005	0.821	0.840	0.859	0.885	0.927	0.932
Mean	0.759	0.777	0.837	0.860	0.835	0.850
Median	0.783	0.802	0.845	0.864	0.841	0.862
Standard Deviation	0.070	0.074	0.021	0.025	0.113	0.102

seven experimental conditions are biological meaningful hits is discussed in detail below (see chapter 3.2.2.3).

For a direct comparison of the significant nodes obtained by the three methods, the pairwise intersections and set unions were determined and, based on them, the Jaccard indices ($J(A,B) = \frac{|A \cap B|}{|A \cup B|}$) were computed to get normalized measures for the relations of both mentioned sets, respectively. Considering this measure, the KS-test and the U-test showed the highest pairwise average similarity, followed by the combinations t-test/U-test and, finally, KS-test/t-test (Tab. 7). The difference between the Jaccard indices was substantial between the first and the third combination (p-value of 0.01 in a two-sided Student's t-test).

In order to expand the comparison of the three threshold-free methods beyond the significant nodes and, thus, to be independent of the chosen significance level and method of correction for multiple testing, the pairwise correlations were computed for each data set of Tab. 6. Interestingly, according to the Pearson's correlation coefficients (Tab. 8), the U-test showed the highest correlation with the other two tests. The corresponding coefficients were both in the same range with an overall high average value of greater than 0.83 (r_{mean} of 0.837 for the KS-test and 0.835 for the t-test). In contrast, the t- and the KS-test bore a moderate but clearly smaller average correlation of around 0.75. The correlation coefficients were all proven to be significantly different from zero (one-tailed t-test). In addition, the medium-sized differences of nearly 0.1 between the means of the former two correlation coefficients and the last one were significant (two-tailed paired

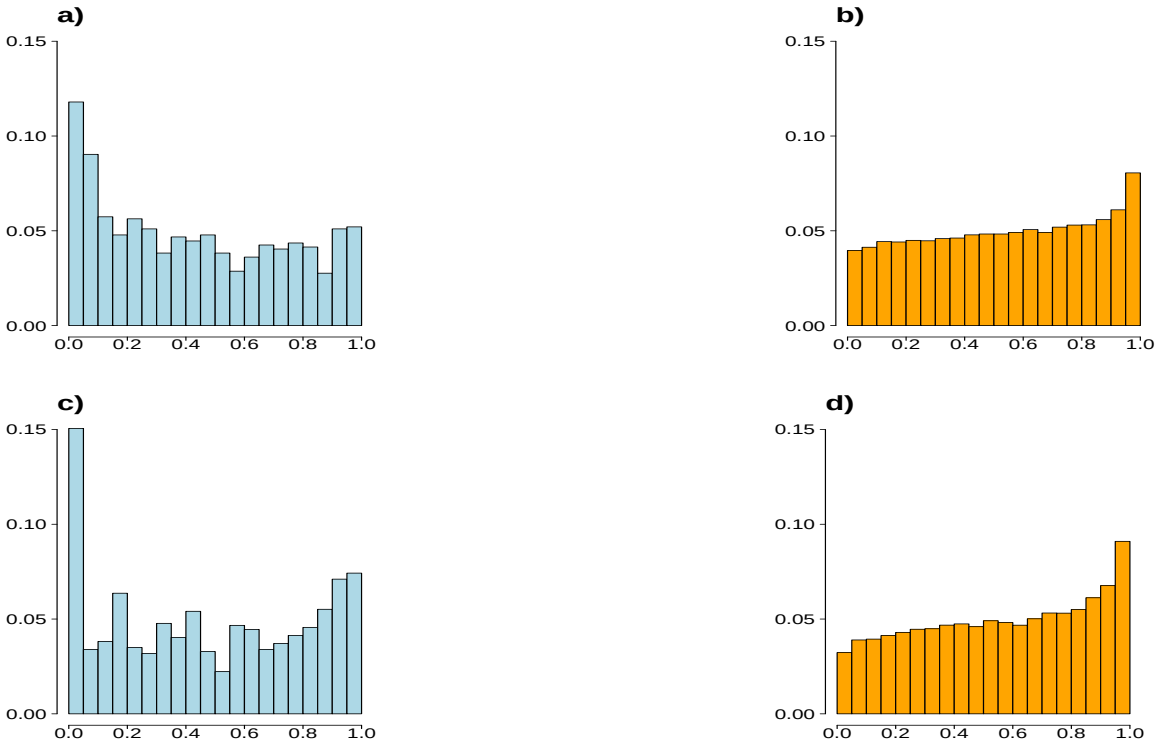


Figure 19: Change of the frequency distribution of the GO nodes' p-values after random permutation of the analyzed gene expression matrix.
The Y-axis reflects the proportion of nodes to which p-values of a specific interval (X-axis, interval size 0.05) are assigned. *lrp*-knockout data set from Hung *et al.*, 2002 with a) original values and b) randomized gene expression values and O_2-tension data set from Salmon *et al.*, 2003 with c) original values and d) randomized gene expression values (see Tab. 6 for details on the data sets).

t-test). The Spearman's rank coefficients – which is a distribution-free measure and, therefore, does not require input data that fit a normal distribution, which is the case for the GO nodes' p-values – confirmed these findings in each particular (Tab. 8). In addition, in all three pairwise comparisons they were slightly higher than the respective coefficients according to Pearson for the computation of which a linear correlation is assumed.

Summarizing the statistical comparison of the cut-off free tests, the t-test marks most GO nodes as significant, in some cases more than twice as much as the other two methods, which more often yield nearly the same numbers. In addition, the KS-test and the U-test bear the highest pairwise agreement with respect to the significant nodes according to the Jaccard index measure. A correlation analysis confirms the major similarities of these two methods, considering not only the significant but also the p-values of all other GO nodes, which was consistent with the high general agreement between the U-test and the GSEA variant of the KS-test found for eukaryotic expression data (Breslin *et al.*, 2004). The U-test, in turn, shows a similar high correlation with the t-test. In contrast, the KS- and the t-test have an average smaller correlation.

Finally, prior to the biological assessment and interpretation of the JProGO case study, another data-driven aspect of the threshold-free approaches should be discussed. Can the

non-random phenomena of the gene expression data be distinguished from random phenomena? For this purpose, two of the expression data sets (*lrp* knockout and anaerobic growth of wild type cells, Tab. 6 lines 6 and 7) were selected, and a functional interpretation was performed using a command line (precursor) version of JProGO. In this analysis, both the original *ppde* of the expression matrix and a series of 100 random permutations of the expression levels of these data were employed as input data. Then, the distributions of the p-values that were obtained for all analyzed GO nodes were compiled (see Fig. 19 for KS-test). For both original expression data sets, a peak near a p-value of 0 was found that comprises, amongst others, the significant nodes (Fig. 19a and 19c). In contrast, after the random permutations of the expression values, this peak disappeared and the histogram rather resembled a uniform distribution for the nodes with lower p-values (< 0.4). The p-values of the remaining nodes show a tendency to rise towards a peak close to 1 (Fig. 19b and 19d).

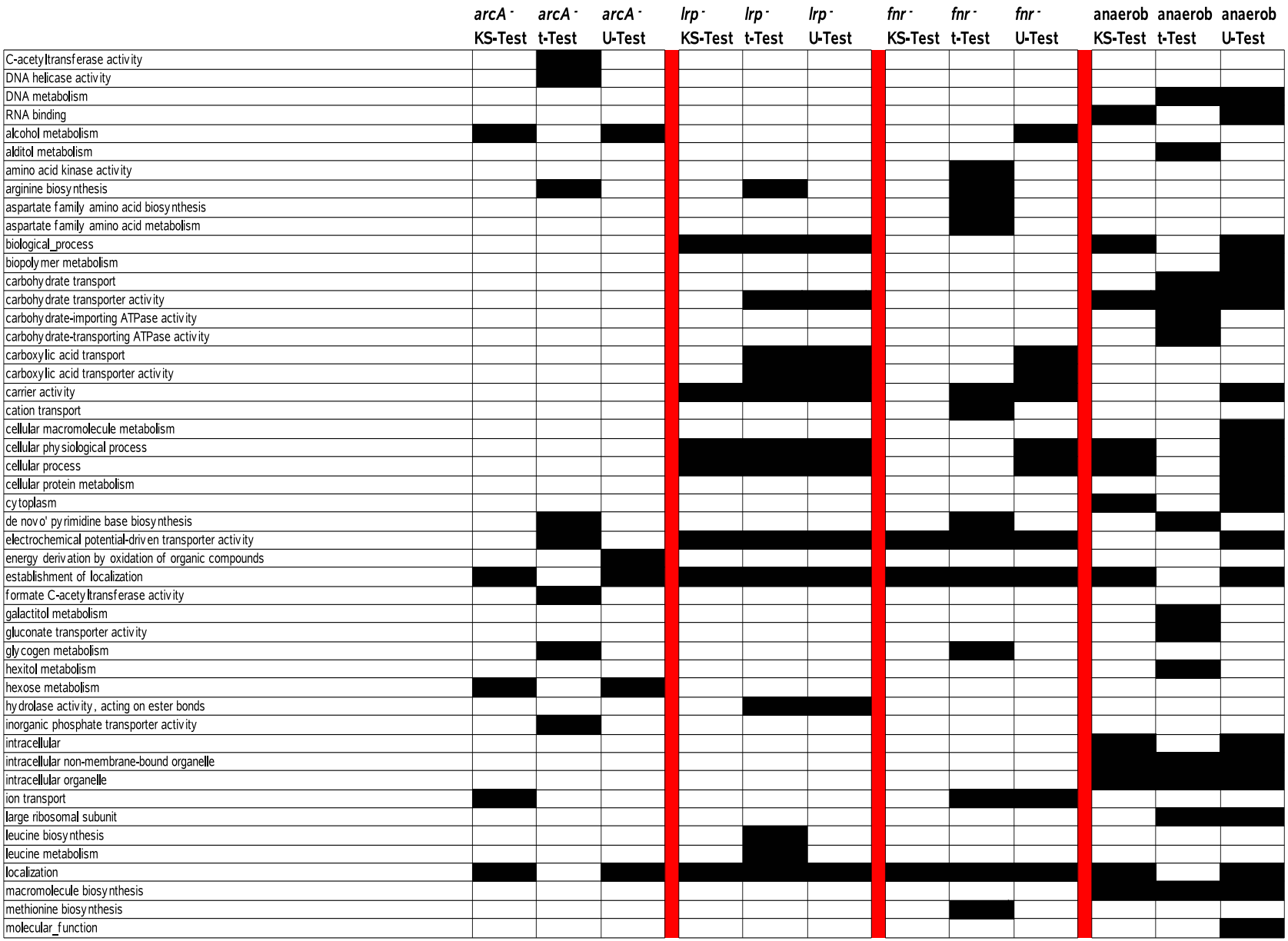
Altogether, the outlined quality check indicates that the application of the KS-test to the analysis of prokaryotic gene expression data results in a reliable p-value distribution. This distribution is due to biological effects since it is strongly perturbed by random permutation of the expression values. Due to the high overall correlation ($R > 0.7$) of the KS-test with the t-test and U-test with respect to the GO nodes' p-values, similarly biological meaningful results are expected for these threshold-free methods.

3.2.2.3 Biological Interpretation and Assessment of the Results

In this chapter the results of the JProGO case study are assessed from a biological point of view. For this purpose the expectations of the authors of the expression data sets (Tab. 6) on the outcomes of the experiments performed by them were incorporated. These expectations were combined with existing biological knowledge and compared to the statistically significant GO nodes computed by JProGO for the three threshold-free methods (Fig. 20 and 21). Each experimental condition (see Tab. 6) is presented below, starting with an inspection of the GO nodes found by all three free methods – the intersection – followed by the nodes that are only identified by one or two of them.

Anaerobic growth

Upon switching from aerobic to anaerobic growth conditions, a major alteration in the central energy consuming and producing biochemical processes was expected for the *E. coli* K-12 wild type strain (Spiro and Guest, 1991; Kang *et al.*, 2005). When no alternative electron acceptor is available under anaerobic conditions, which was the case for the two corresponding data sets (Fig. 20, col. 10-12 and Fig. 21, col. 5-7), fermentative processes should be affected. The results obtained with JProGO for the data set of Kang *et al.* (2005) clearly met these expectations since the GO nodes '*energy derivation by oxidation of organic compounds*', '*oxidoreductase activity*', '*tricarboxylic acid cycle*', '*aerobic respiration*' and '*acetyl-CoA metabolism*' were marked as significant by all three cut-off free methods (Fig. 21, col. 5-7). Additional nodes biologically meaningful in this context comprise '*succinate dehydrogenase activity*', *NADH dehydrogenase activity*' and'*glycerol-3-phosphate metabolism*' which were only found by the t-test, whereas '*oxidoreductase activity, acting on NADH or NADPH*' was identified by both the t- and U-test. The nodes '*cellular respiration*' and '*quinone binding*' (compare to Malpica *et al.*, 2004) were found by both, the KS-test and U-test. The significant nodes '*ferredoxin hydrogenase complex*' (t-test) and '*nickel ion binding*' (t- and U-test) are also biologically meaningful


arcA - KS-Test
arcA - t-Test
arcA - U-Test
lrp - KS-Test
lrp - t-Test
lrp - U-Test
fnr - KS-Test
fnr - t-Test
fnr - U-Test
anaerob KS-Test
anaerob t-Test
anaerob U-Test
C-acetyltransferase activity
DNA helicase activity
DNA metabolism
RNA binding
alcohol metabolism
alditol metabolism
amino acid kinase activity
arginine biosynthesis
aspartate family amino acid biosynthesis
aspartate family amino acid metabolism
biological_process
biopolymer metabolism
carbohydrate transport
carbohydrate transporter activity
carbohydrate-importing ATPase activity
carbohydrate-transporting ATPase activity
carboxylic acid transport
carboxylic acid transporter activity
carrier activity
cation transport
cellular macromolecule metabolism
cellular physiological process
cellular process
cellular protein metabolism
cytoplasm
de novo' pyrimidine base biosynthesis
electrochemical potential-driven transporter activity
energy derivation by oxidation of organic compounds
establishment of localization
formate C-acetyltransferase activity
galactitol metabolism
gluconate transporter activity
glycogen metabolism
hexitol metabolism
hexose metabolism
hydrolase activity, acting on ester bonds
inorganic phosphate transporter activity
intracellular
intracellular non-membrane-bound organelle
intracellular organelle
ion transport
large ribosomal subunit
leucine biosynthesis
leucine metabolism
localization
macromolecule biosynthesis
methionine biosynthesis
molecular_function

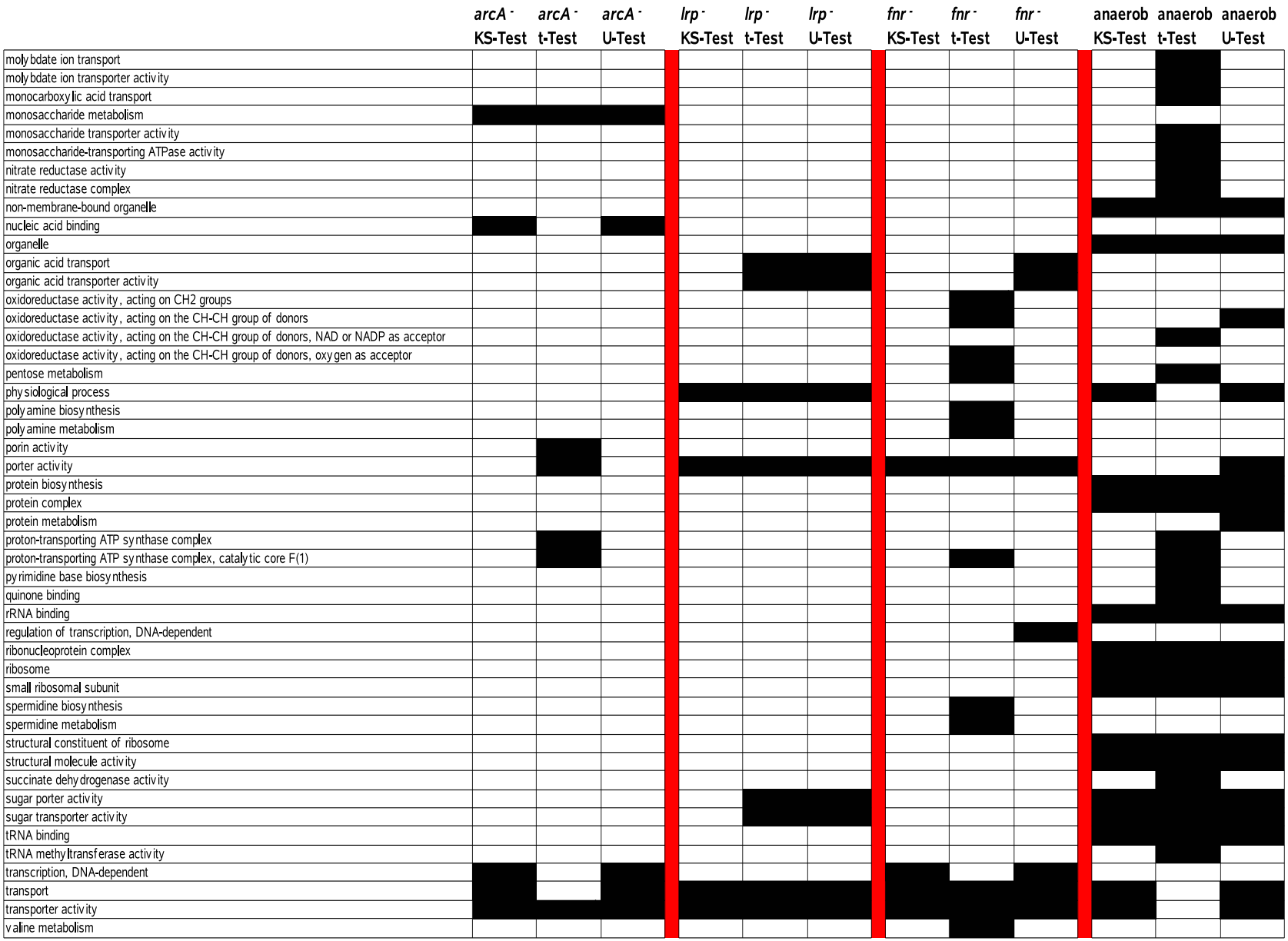

arcA - KS-Test
arcA - t-Test
arcA - U-Test
lrp - KS-Test
lrp - t-Test
lrp - U-Test
fnr - KS-Test
fnr - t-Test
fnr - U-Test
anaerob KS-Test
anaerob t-Test
anaerob U-Test
molybdate ion transport
molybdate ion transporter activity
monocarboxylic acid transport
monosaccharide metabolism
monosaccharide transporter activity
monosaccharide-transporting ATPase activity
nitrate reductase activity
nitrate reductase complex
non-membrane-bound organelle
nucleic acid binding
organelle
organic acid transport
organic acid transporter activity
oxidoreductase activity, acting on CH2 groups
oxidoreductase activity, acting on the CH-CH group of donors
oxidoreductase activity, acting on the CH-CH group of donors, NAD or NADP as acceptor
oxidoreductase activity, acting on the CH-CH group of donors, oxygen as acceptor
pentose metabolism
physiological process
polyamine biosynthesis
polyamine metabolism
porin activity
porter activity
protein biosynthesis
protein complex
protein metabolism
proton-transporting ATP synthase complex
proton-transporting ATP synthase complex, catalytic core F(1)
pyrimidine base biosynthesis
quinone binding
rRNA binding
regulation of transcription, DNA-dependent
ribonucleoprotein complex
ribosome
small ribosomal subunit
spermidine biosynthesis
spermidine metabolism
structural constituent of ribosome
structural molecule activity
succinate dehydrogenase activity
sugar porter activity
sugar transporter activity
tRNA binding
tRNA methyltransferase activity
transcription, DNA-dependent
transport
transporter activity
valine metabolism

Figure 20: JProGO case study (first part) performed with preprocessed expression data from *E. coli*. Significant GO nodes obtained for the three threshold-free methods are shown (black rectangles) for the four conditions *arcA* knockout (columns 1-3), *lrp* knockout (columns 4-6), *fnr* knockout (columns 7-9) and wild type cells grown anaerobically (columns 10-12). In each case aerobically grown wild type cells served as control condition. The probabilities of differential expression (ppde) provided by the authors of the studies (Hung *et al.*, 2002; Salmon *et al.*, 2003, 2005) were used as input data.

in this context since hydrogenase activity is typical for anaerobic energy metabolism and nickel is a cofactor of the hydrogenase.

The second data set of Salmon *et al.* (2003) yielded as intersection in most cases more general nodes, being closer to the root node of the GO graph, than the data set of Kang *et al.* (2005). For example, '*protein biosynthesis*', '*ribosome*', '*sugar porter activity*' and '*macromolecule biosynthesis*' are found by all three statistical tests and meet the expectation (Fig. 20, col. 11-13). In addition, '*oxidoreductase activity, acting on CH-CH groups*' and a child node of it were identified by the t-test and U-test. The t-test also found '*succinate dehydrogenase activity*' and '*quinone binding*' (see data set of Kang *et al.* (2005)) and marked '*nitrate reductase activity*' and '*nitrate reductase complex*' as significant. Whereas the above mentioned nodes seemed to fit the expectation for a change from aerobic to anaerobic growth, this in principle also the case for the functions '*molybdate ion transporter activity*' (molybdate is a cofactor of the nitrate reductase complex) and '*tRNA methyltransferase activity*' (part of the protein biosynthesis), which were also found by the t-test.

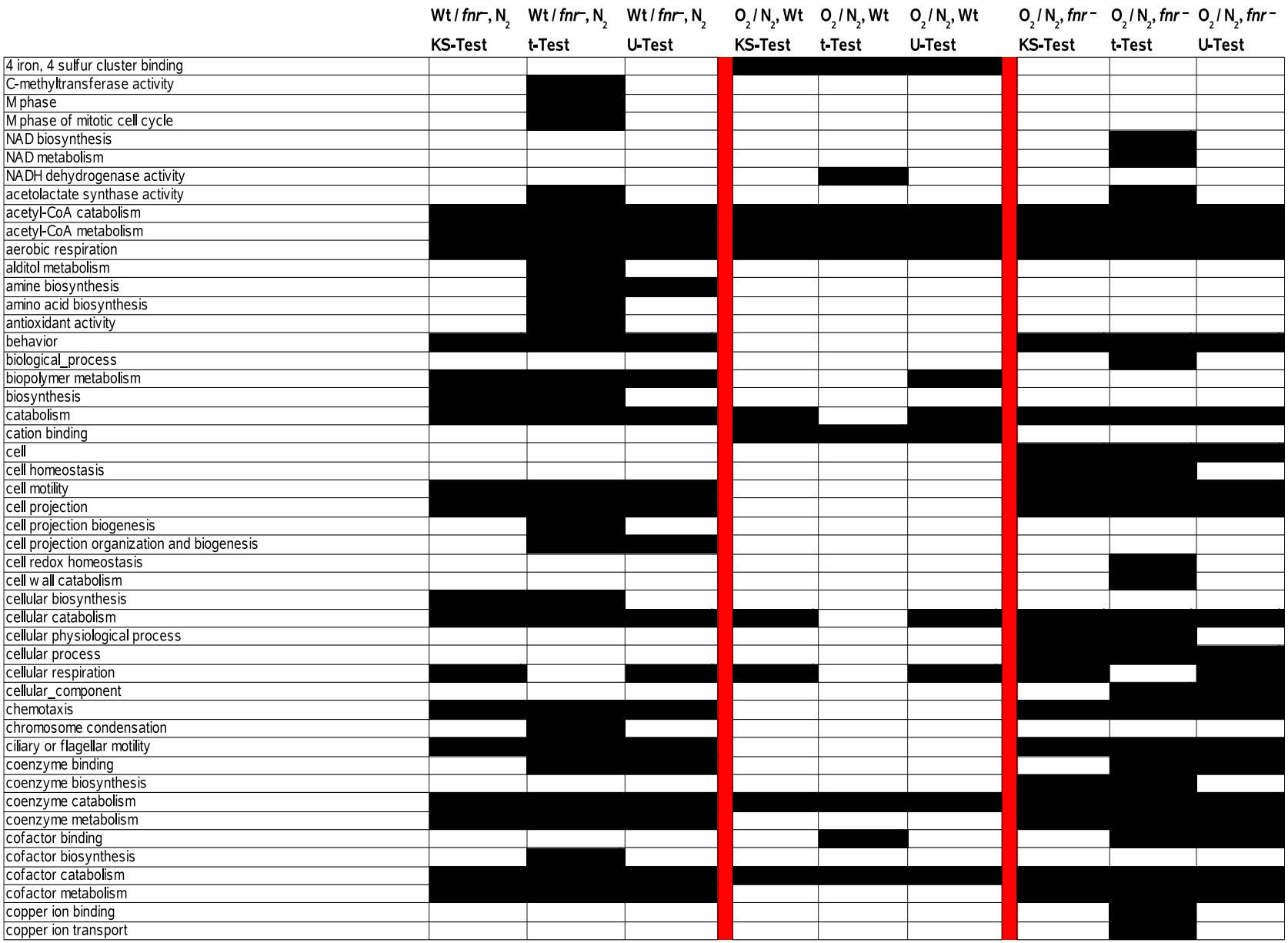
Knockout of the *arcA* gene

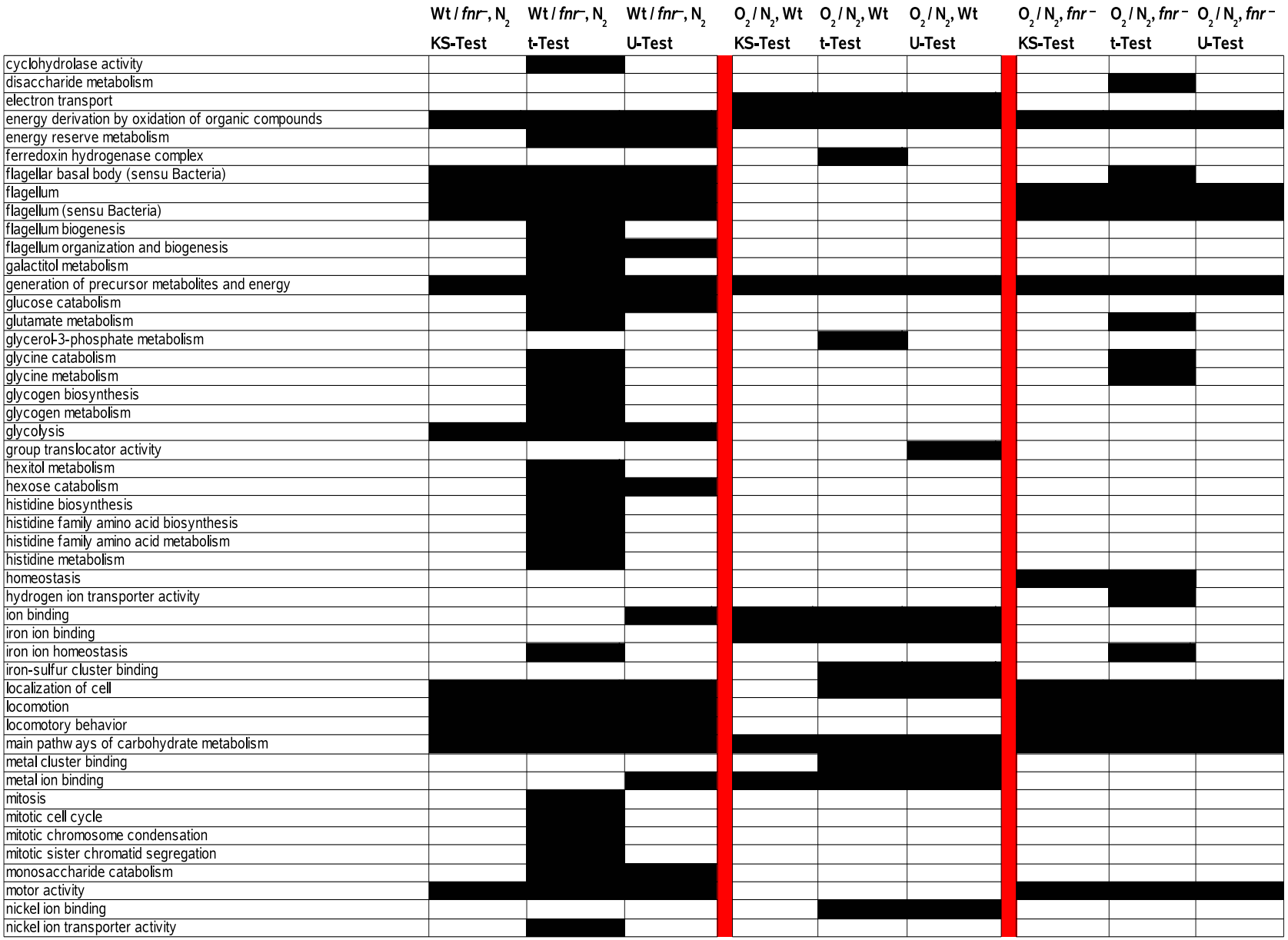
One of the main transcriptional sub-networks in response to oxygen depletion is controlled by the global regulator ArcA, which is part of the ArcAB two-component system. For the *arcA* knockout data set of Salmon *et al.* (2005), all three cut-off free methods found GO nodes, which fit with the biological expectation of *arcA* mutant cells, e.g. '*monosaccharide metabolism*' and '*transporter activity*'. The KS- and U-test further specified the process '*monosaccharide metabolism*' by marking '*hexose metabolism*' as significant. The GO node '*energy derivation by oxidation of organic compounds*', which was found by the U-test, reflected the profound ArcA-mediated change in energy production under anaerobic conditions. Furthermore, the biological process '*transcription, DNA-dependent*', which was identified by both the KS-test and U-test, might indicate a broad change in the transcriptional program of the cell that was caused by the global regulator ArcA and also may affect the activities of other downstream regulators. The changes in the expression pattern of the genes belonging to the '*glycogen metabolism*', which was marked as significant by the t-test, seemed to be a physiological reaction of the cells, since an accumulation of this polyglucan was observed before under limited growth conditions (e.g. anaerobic conditions) in the presence of a rich carbon source (Preiss and Romeo, 1994). Altogether, a small number of nodes was identified by the three methods for this knockout condition and, obviously, several initially expected functions and processes were missing.

Knockout of the *fnr* gene

Besides the two-component system response regulator ArcA, Fnr is the other main tran-

scriptional regulator that controls the transition from aerobic to anaerobic growth in *E. coli* (Spiro and Guest, 1991; Salmon *et al.*, 2003; Kang *et al.*, 2005). The JProGO-based analysis of the data set of Kang *et al.* (2005) reflects the fundamental alteration of the transcriptional program in an *fnr* knockout strain under anaerobic conditions: All three statistical tests concordantly identified significant changes in the '*oxidoreductase activity*' in general and '*main pathways of carbohydrate metabolism*' comprising the '*glycolysis*' and the '*tricarboxylic acid cycle*' as well as other processes of the '*aerobic respiration*'. Further common statistically significant nodes were '*acetyl-CoA catabolism*', '*energy derivation by oxidation of organic compounds*' that could, at least partially, be traced back to the identification of their more specific child nodes ('*glycolysis*' and '*tricarboxylic acid cycle*'). In addition, also the motility of the cell as represented by the nodes '*flagellum (sensu Bacteria)*', '*ciliary and flagellar motility*' and '*chemotaxis*' seemed be affected by Fnr. Besides these nodes, that meet the expectation in this context and that were commonly identified by all three methods, the t-test found several additional GO nodes: Whereas, for example, '*succinate dehydrogenase activity*', '*porphyrin biosynthesis*', '*iron ion homeostasis*' and '*glycogen metabolism*' (see also *arcA* knockout) fit well with the expectation, it is rather not the case for '*cyclohydrolase activity*', '*histidine metabolism*' or '*serine family amino acid catabolism*'. '*Metal ion binding*' was another meaningful node only found by the U-test.


Wt / fnr⁻, N2 KS-Test
Wt / fnr⁻, N2 t-Test
Wt / fnr⁻, N2 U-Test
O2 / N2, Wt KS-Test
O2 / N2, Wt t-Test
O2 / N2, Wt U-Test
O2 / N2, fnr⁻ KS-Test
O2 / N2, fnr⁻ t-Test
O2 / N2, fnr⁻ U-Test
4 iron, 4 sulfur cluster binding
C-methyltransferase activity
M phase
M phase of mitotic cell cycle
NAD biosynthesis
NAD metabolism
NADH dehydrogenase activity
acetolactate synthase activity
acetyl-CoA catabolism
acetyl-CoA metabolism
aerobic respiration
alditol metabolism
amine biosynthesis
amino acid biosynthesis
antioxidant activity
behavior
biological_process
biopolymer metabolism
biosynthesis
catabolism
cation binding
cell
cell homeostasis
cell motility
cell projection
cell projection biogenesis
cell projection organization and biogenesis
cell redox homeostasis
cell wall catabolism
cellular biosynthesis
cellular catabolism
cellular physiological process
cellular process
cellular respiration
cellular_component
chemotaxis
chromosome condensation
ciliary or flagellar motility
coenzyme binding
coenzyme biosynthesis
coenzyme catabolism
coenzyme metabolism
cofactor binding
cofactor biosynthesis
cofactor catabolism
cofactor metabolism
copper ion binding
copper ion transport


Wt / fnr⁻, N2 KS-Test
Wt / fnr⁻, N2 t-Test
Wt / fnr⁻, N2 U-Test
O2 / N2, Wt KS-Test
O2 / N2, Wt t-Test
O2 / N2, Wt U-Test
O2 / N2, fnr⁻ KS-Test
O2 / N2, fnr⁻ t-Test
O2 / N2, fnr⁻ U-Test
cyclohydrolase activity
disaccharide metabolism
electron transport
energy derivation by oxidation of organic compounds
energy reserve metabolism
ferredoxin hydrogenase complex
flagellar basal body (sensu Bacteria)
flagellum
flagellum (sensu Bacteria)
flagellum biogenesis
flagellum organization and biogenesis
galactitol metabolism
generation of precursor metabolites and energy
glucose catabolism
glutamate metabolism
glycerol-3-phosphate metabolism
glycine catabolism
glycine metabolism
glycogen biosynthesis
glycogen metabolism
glycolysis
group translocator activity
hexitol metabolism
hexose catabolism
histidine biosynthesis
histidine family amino acid biosynthesis
histidine family amino acid metabolism
histidine metabolism
homeostasis
hydrogen ion transporter activity
ion binding
iron ion binding
iron ion homeostasis
iron-sulfur cluster binding
localization of cell
locomotion
locomotory behavior
main pathways of carbohydrate metabolism
metal cluster binding
metal ion binding
mitosis
mitotic cell cycle
mitotic chromosome condensation
mitotic sister chromatid segregation
monosaccharide catabolism
motor activity
nickel ion binding
nickel ion transporter activity

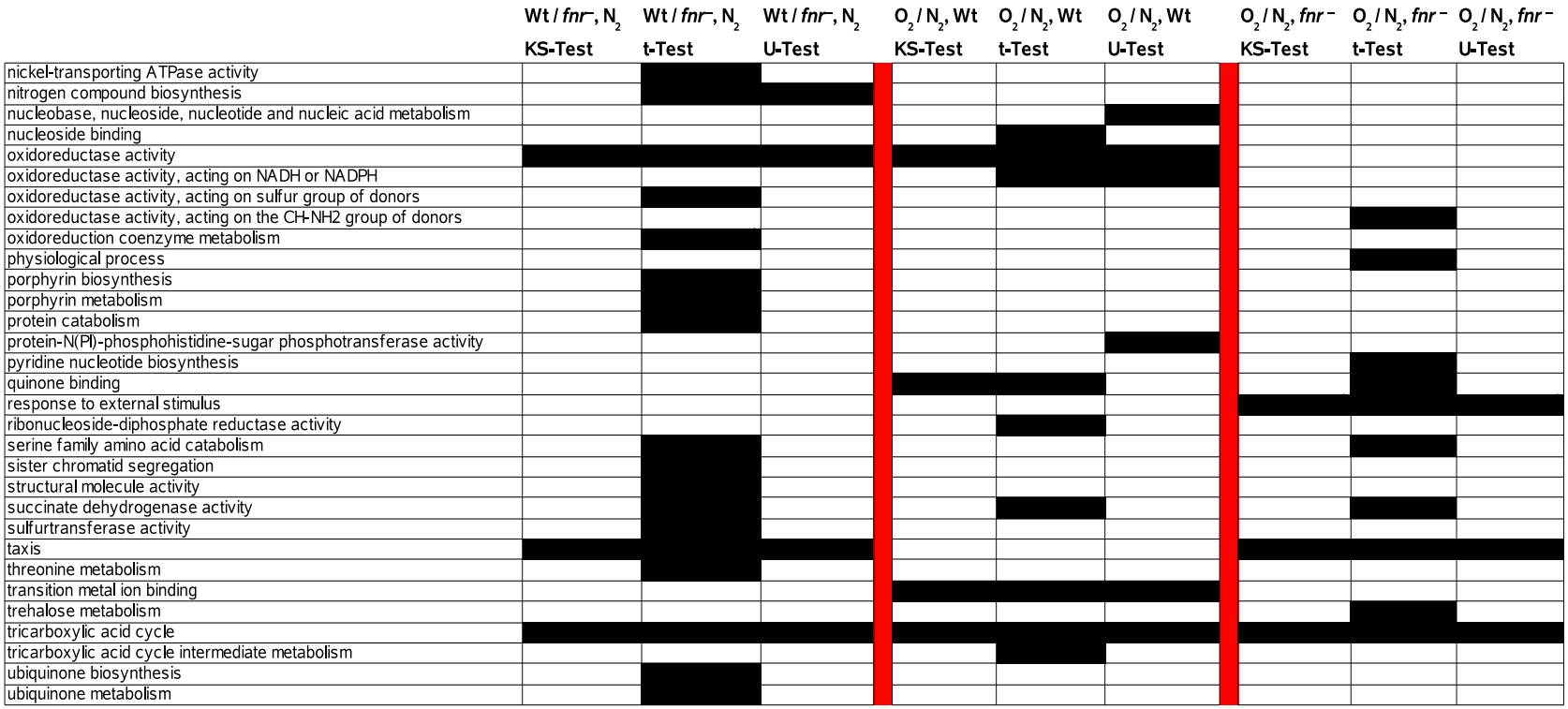

Wt / fnr⁻, N2 KS-Test
Wt / fnr⁻, N2 t-Test
Wt / fnr⁻, N2 U-Test
O2 / N2, Wt KS-Test
O2 / N2, Wt t-Test
O2 / N2, Wt U-Test
O2 / N2, fnr⁻ KS-Test
O2 / N2, fnr⁻ t-Test
O2 / N2, fnr⁻ U-Test
nickel-transporting ATPase activity
nitrogen compound biosynthesis
nucleobase, nucleoside, nucleotide and nucleic acid metabolism
nucleoside binding
oxidoreductase activity
oxidoreductase activity, acting on NADH or NADPH
oxidoreductase activity, acting on sulfur group of donors
oxidoreductase activity, acting on the CH-NH2 group of donors
oxidoreduction coenzyme metabolism
physiological process
porphyrin biosynthesis
porphyrin metabolism
protein catabolism
protein-N(PI)-phosphohistidine-sugar phosphotransferase activity
pyridine nucleotide biosynthesis
quinone binding
response to external stimulus
ribonucleoside-diphosphate reductase activity
serine family amino acid catabolism
sister chromatid segregation
structural molecule activity
succinate dehydrogenase activity
sulfurtransferase activity
taxis
threonine metabolism
transition metal ion binding
trehalose metabolism
tricarboxylic acid cycle
tricarboxylic acid cycle intermediate metabolism
ubiquinone biosynthesis
ubiquinone metabolism

Figure 21: JProGO case study (second part) performed with preprocessed expression data from *E. coli*: The significant GO nodes obtained for the three threshold-free methods are shown for the four conditions *fnr* knockout versus wildtype (aerob), *fnr* knockout versus wildtype (anaerob), *fnr* knockout aerob versus anaerob and wild type cells aerob versus anaerob. For further information, refer to Fig. 20.

For the second data set that compares an *fnr* knockout strain with the wild type strain (Salmon *et al.*, 2003) the three methods mark either an equally small number (KS-test) or a considerably smaller (t- and U-test) number of nodes as significant (Tab. 7). Expected nodes that were found by all three methods comprise '*electrochemical potential-driven transporter activity*' and '*transport*'. In addition, the t-test identified several forms of '*oxidoreductase activity*' such as '*oxidoreductase activity, acting on CH2 groups*', '*oxidoreductase activity, acting on CH-CH group of donors, oxygen as acceptor*' as well as '*proton-transporting ATP synthase complex, catalytic core F(1)*' and '*glycogen metabolism*'. These nodes also fit the expectation, in contrast to some other nodes e.g. '*de novo pyrimidine base synthesis*', '*arginine biosynthesis*' and '*spermidine biosynthesis*'.

The small number of significant nodes found by the three methods may partially be due to the weaker difference between the two conditions in the second gene expression data set and the comparatively stringent level of significance chosen (FDR of 0.05). For example, for all three methods the expected process '*monosaccharide metabolism*' was followed not until 15 nodes after the last significant one in the list of all nodes sorted by their p-values.

Another two-condition comparison was the fnr^- strain, grown aerobically versus anaerobically (Kang *et al.*, 2005). In this case, predominantly Fnr-independent adaptations to the anaerobic growth conditions were expected (see Kang *et al.*, 2005). However, the observed alterations were similar to those of the comparison between the mutant and wild type strain under anaerobic conditions. This includes, for example, '*acetyl-CoA catabolism*', '*aerobic respiration*', '*energy derivation by oxidation of organic compounds*', as well as '*ciliary and flagellar motility*' and '*chemotaxis*'. Interestingly, in contrast to the comparison between the fnr^- mutant and the wild type, the '*glycolysis*' was not identified. This could reflect a prominent role of Fnr in the regulation of this biological process.

The last analyzed data set referring to the knockout of the *fnr* gene investigated the difference between the wild type and the fnr^- strain under aerobic growth conditions (see Tab. 6, line 5). Since Fnr with its O_2-sensitive $[4Fe-4S]^{2+}$ cluster is only active under anaerobic growth conditions, no broad transcriptional changes were expected for this data set. Therefore, it was regarded as a *null control*. Indeed, none of the three cut-off free methods found any significant node (see chapter 3.2.2.2).

Knockout of the *lrp* gene

Like Fnr and ArcA, the leucine responsive regulatory protein Lrp is a global transcriptional regulator of *E. coli*. It is involved in modulating a variety of metabolic functions, including the catabolism and anabolism of amino acids as well as pili synthesis (Brinkman *et al.*, 2003). Therefore, the Lrp regulon comprises a lot of genes that are responsible for amino acid metabolism, and a lower number of genes that are involved in pili synthesis (Hung *et al.*, 2002; Brinkman *et al.*, 2003). For the *lrp* knockout data set of Hung *et al.*

(2002) all three cut-off free methods found GO nodes like '*porter activity*' and '*transport*', which comply with the expected elevation of transporting small organic molecules such as amino acids. In this context, the t- and U-test additionally identified '*carboxylic acid transport*' and '*sugar porter activity*'. Furthermore, with the '*leucine biosynthesis*' and the '*arginine biosynthesis*' the t-test marked two nodes as significant that directly represent the anabolism of selected amino acids. Especially for the '*leucine biosynthesis*' this fits well with the biological expectation. Similar to the analysis results of the *fnr*$^-$ data set of Salmon *et al.* (2003), considering nodes with a p-value slightly higher than that of the last significant node would lead to the inclusion of further biological meaningful functions which might be due to the comparatively stringent level of significance (FDR of 0.05). Indeed, when increasing it to a quite liberal value of 0.15, the t- and U-test additionally found '*histidine biosynthesis*' and '*histidine family amino acid biosynthesis*'.

Concluding remarks
The biological assessment and comparison of the three cut-off free methods showed that all these methods have the potential to identify GO nodes that are consistent with the biological expectation of the analyzed experimental conditions. In all cases, a subset of these nodes was in the intersection of all three approaches. However, the methods differed in the number of expected nodes found which – maybe coincidentally – corresponds to the absolute number of statistically significant nodes (see Tab. 7): Thus, the KS-test, which generally marks the lowest overall number of nodes as significant, identified the fewest biologically reasonable nodes and, vice versa, the t-test, which marked the highest number of nodes as significant, found most biologically meaningful nodes, while the results of the U-test were in-between. Despite the fact that the t-test finds most reasonable nodes, it should be mentioned that it, additionally, identifies in each case several nodes which do not match well with the expectation. Thus, the U-test might constitute a good compromise. Another approach, based on the results of the methods presented, would be to focus on those GO nodes, which are marked as significant by at least two methods, e.g. the t- and the U-test.

Independently of the employed method of analysis, yet another factor influences the extent and number of meaningful GO nodes that were identified: the analyzed data set itself. Whereas for the three data sets of Kang *et al.* (2005), which provide p-values, the coincidence with the biological expectation was high, it was lower for the other three data sets, which provided ppde. The latter three indicated the important role of the chosen level of significance and relaxing it led to the inclusion of more biological meaningful nodes for the t- and U-test.

3.2.2.4 Influence of the Type of Expression Data: Ratios versus Test Statistics

For all hitherto analyses of the case study presented above, the test statistics, p-values or ppde, were taken as input data. It is well known that test statistics such as the ppde generated by a regularized t-test do, in contrast to expression ratios, not only take the average expression levels of all replicate measurements into account but also their variances (see Baldi and Long, 2001; Hatfield *et al.*, 2003). Therefore, using test statistics constitutes an advantage because they are not as susceptible to experimental noise as the expression ratios. However, vice versa, expression ratios directly reflect the relative changes in the expression level of a gene, and this information, which can

Table 9: Rank correlation of the two different types of expression data: ratios versus test statistics (ppde, p-values). The data sets are those of the *E. coli* case study shown in Tab. 6.

Conditions & Reference	$R_{Spearman}$
arcA$^-$ vs. Wt (anaerob); Salmon *et al.* (2005)	0.923
fnr$^-$ vs. Wt (anaerob); Salmon *et al.* (2003)	0.909
fnr$^-$ vs. Wt (anaerob); Kang *et al.* (2005)	0.766
fnr$^-$, aerob vs.*fnr*$^-$, anaerob; Kang *et al.* (2005)	0.771
fnr$^-$ vs. Wt (aerob); Kang *et al.* (2005)	0.772
lrp$^-$ vs. Wt; Hung *et al.* (2002)	0.858
Wt, aerob vs. Wt, anaerob; Salmon *et al.* (2003)	0.873
Wt, aerob vs. Wt, anaerob; Kang *et al.* (2005)	0.789
Mean	0.833
Median	0.823
Standard Deviation	0.065

become biologically relevant, is lost when solely using test statistics. For example, in an expression profiling experiment genes might exist that have a high probability of differential expression (e.g. ppde), but, at the same time, show only a slight change in their relative expression level, e.g. only 1.2 fold. Clearly, such genes are differentially expressed from a statistical point of view, but the biological relevance of this change might be minor (see Bickel, 2004). Therefore, due to their intuitive comprehensibility and due to the lack of a sufficient number of replicates especially at the beginning of expression profiling experiments a few years ago, experimental biologists preferred and nowadays sometimes still prefer the use of expression ratios to test statistics. This was the motivation for an expansion of the JProGO case study towards expression ratios. In this context, first of all, the differences between expression ratios and test statistics for each analyzed data set were quantified, since for the reasons outlined above, they were expected to assign different ranks to the genes. For this purpose, the rank correlation coefficients were determined for the expression data sets, whereas for all ratios below one their reciprocals were computed (Tab. 9). The overall rank correlation was high, ranging from 0.77 to 0.93 (Tab. 9). Thus, both types of expression data were not too different with respect to the ranks of the genes. To which extent this effect is propagated to the results of a high-level analysis was investigated in the following. For this purpose, the expression ratios were taken from the 8 expression data sets of the case study (Tab. 6) as input data. For this analysis, as for the test statistics (chapter 3.2.2.2), the U-test was chosen, since it constituted a good compromise between the KS-test, which found least, and the t-test, which found most and most meaningful GO nodes (chapter 3.2.2.2 and 3.2.2.3). All other parameters were kept the same as described above (chapter 3.2.2.2). The compilation of all significant GO nodes (Fig. 22) revealed that for the expression ratios in all cases a higher number of significant hits was obtained (Tab. 10) compared to those obtained with test statistics (Tab. 7). This observation was also made for several other microarray data sets (not shown) and seems to be a general tendency.

Furthermore, as expected a low to moderate correlation between the ratios and the corresponding test statistics was detected over all GO nodes: The rank correlation range from a low value of 0.044 to a medium value of 0.407, whereas the Pearson's correlation

Table 10: Comparison of the JProGO results obtained for the expression ratios with those obtained for the test statistics. Firstly, the total numbers of significant GO nodes obtained for the expression ratios with the U-test (Fig. 22) are shown (col. '#N'). Furthermore, for the comparison of these nodes with those obtained for the test statistics, that are ppde and p-values (Fig. 20 and 21), the Jaccard indices (col. "Jacc.") and the respective absolute (col. "#∩") and relative sizes of the intersections (cols. "$\frac{\#\cap}{\#pval.}$", "$\frac{\#\cap}{\#ratios}$") are listed. Secondly, the Pearson's (col. "r") and the Spearman's rank (col. "R") correlation coefficients of the p-values of all GO nodes computed for the expression ratios and the test statistics – each for the same microarray data set – are given. The data sets analyzed refer to the *E. coli* case study shown in Tab. 6

.

Conditions & Reference	#N	#∩	Jacc.	$\frac{\#\cap}{\#ratios}$	$\frac{\#\cap}{\#pval.}$	r	R
arcA$^-$ vs. Wt (anaerob) Salmon et al., 2005	50	3	0.053	0.060	0.300	0.188	0.222
fnr$^-$ vs. Wt (anaerob) Salmon et al., 2003	56	5	0.074	0.089	0.294	0.359	0.400
fnr$^-$ vs. Wt (anaerob) Kang et al., 2005	58	31	0.456	0.534	0.756	0.344	0.407
fnr$^-$, aerob vs.*fnr*$^-$, anaerob Kang et al., 2005	44	20	0.357	0.455	0.625	0.067	0.143
fnr$^-$ vs. Wt (aerob) Kang et al., 2005	20	0	0.000	0.000	–	0.041	0.042
lrp$^-$ vs. Wt Hung et al., 2002	60	10	0.145	0.167	0.526	0.023	0.070
Wt, aerob vs. Wt, anaerob Salmon et al., 2003	68	25	0.301	0.368	0.625	0.159	0.248
Wt, aerob vs. Wt, anaerob Kang et al., 2005	62	7	0.084	0.113	0.250	0.000	0.044
Mean			0.184	0.223	0.422	0.148	0.197
Median			0.115	0.140	0.413	0.113	0.183
Standard Deviation			0.166	0.200	0.252	0.142	0.149

was always slightly worse (Tab. 10). Interestingly, the correlation was highest for the *arcA*$^-$ and both *fnr*$^-$ knockout data sets, each under anaerobic conditions. When focusing on the significant nodes and their intersections, as represented by the Jaccard indices (Tab. 10), a slightly different picture was obtained: the Jaccard index ranged from 0.00 to 0.45, but did not directly correspond to the correlation coefficient. In all cases, a remarkably high fraction of 25% to 75% of significant nodes were identified with both the test statistics and the expression ratios.

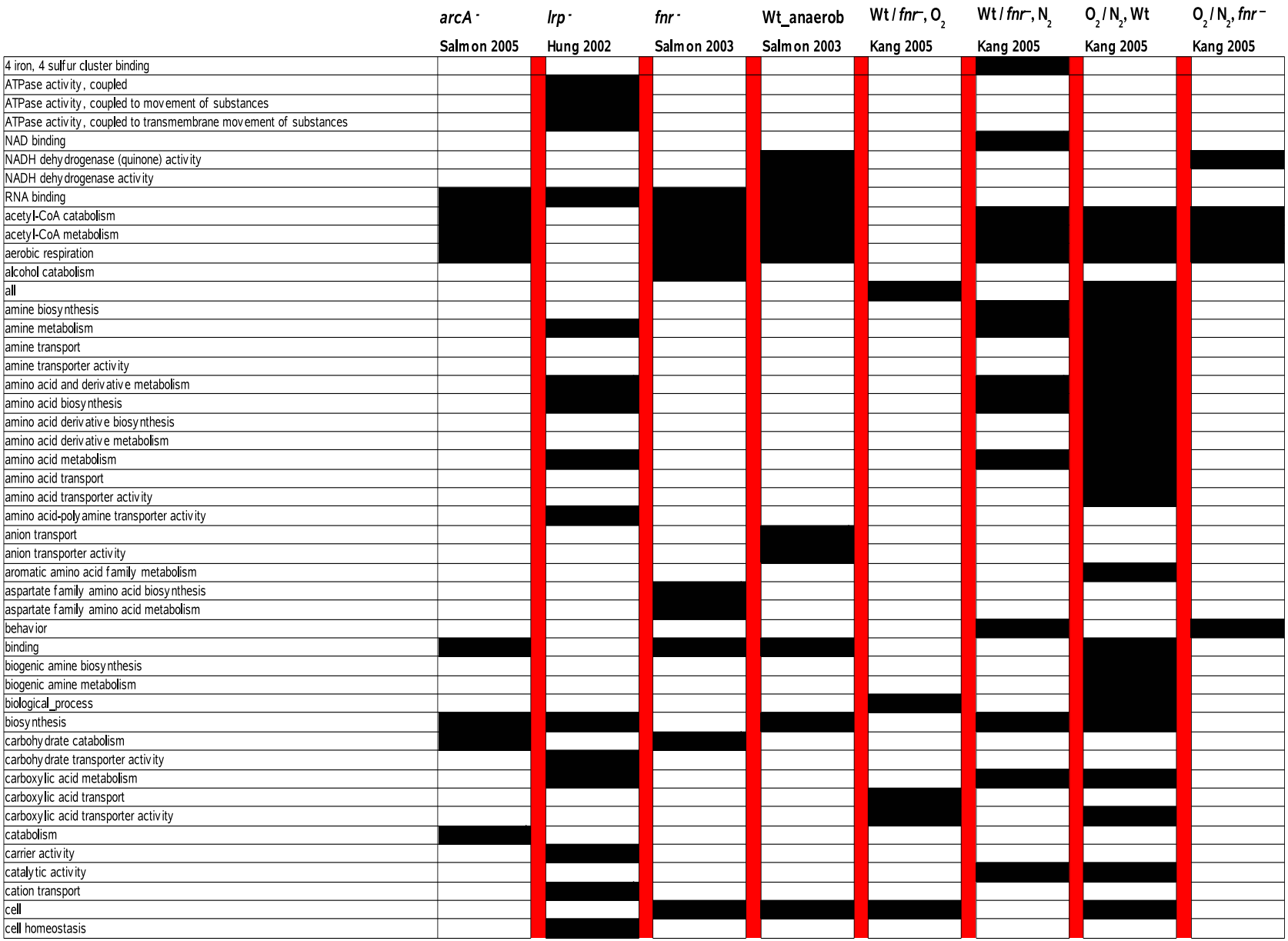
Besides the descriptive statistical comparison of both types of expression data, a short biological assessment of the JProGO-based results is given in the following. Here, the focus is on those nodes found for both types of expression data and those found only for the ratios, and whether these nodes match the biological expectation.

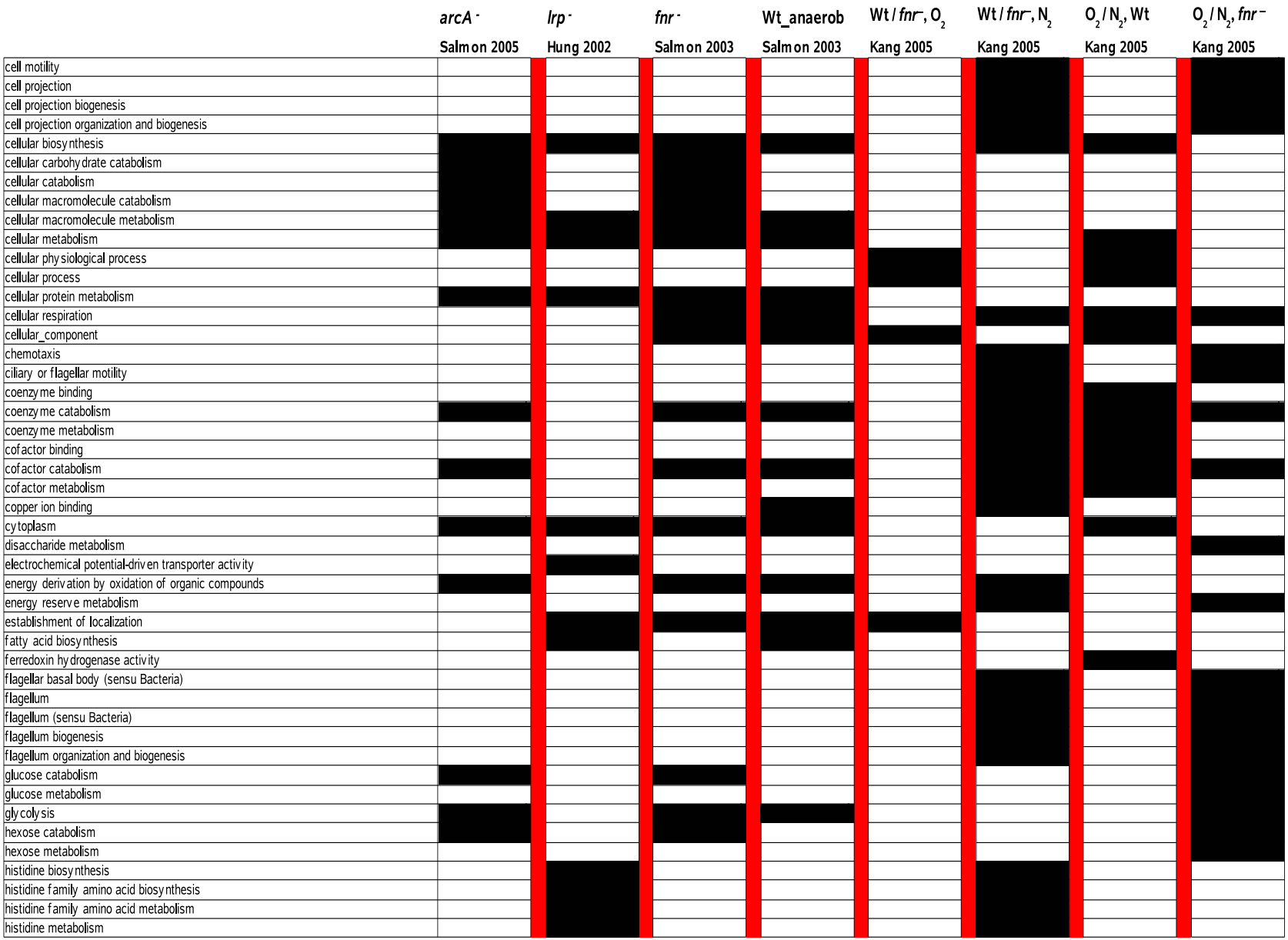
For the *arcA*$^-$ data set only 3 common nodes were found, which represent the ArcA-mediated changes in the '*energy derivation*' and the general changes in the '*DNA-dependent transcription*' caused by the knockout of this global transcriptional regulator. Using the expression ratios, additional biological meaningful nodes such as '*acetyl-CoA metabolism*', '*aerobic respiration*', '*glycolysis*' and '*tricarboxylic acid cycle*' were found. Other nodes that were not primarily expected, such as '*purine nucleotide biosynthesis*', '*protein biosynthesis*' and '*ribosome*', might either be due to unintended differences in the growth behavior of the two strains or attributed to non-biological effects such as experimental noise.

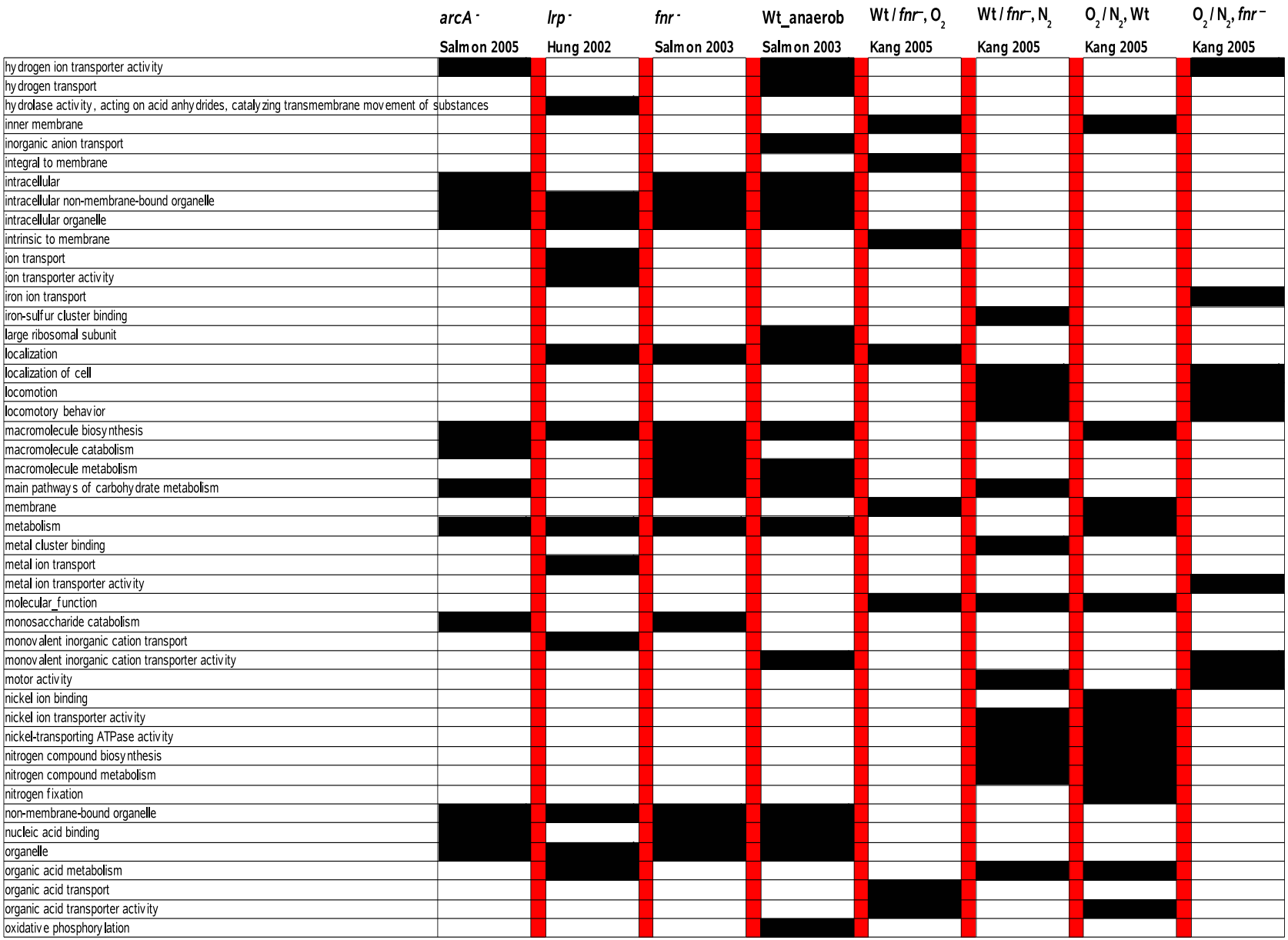
For the *fnr*$^-$ data set of Salmon *et al.* (2003) the only 5 common nodes comprise rather general functions and processes such as '*transporter activity*' and, again, '*transcription, DNA-dependent*'. Similar biological meaningful GO nodes in this context, as for the *arcA*$^-$ knockout data set, were identified when using the expression ratios, for example '*aerobic respiration*', '*glycolysis*' and '*tricarboxylic acid cycle*' (see above). For the other *fnr* knockout data set (Kang *et al.*, 2005) tested under anaerobic conditions, due to the larger overlap between both types of expression data (Jaccard index of almost 0.5), the set of common nodes contained many expected ones (see Fig. 22): for example, nodes concerning the change from aerobic to anaerobic metabolism such as those only found with the ratios for the other two data sets ('*aerobic respiration*' and other, see above). Whether the '*flagellum*' and related nodes, which were marked as significant, were really affected under these conditions is at least questionable. Another point was that the overlap of significant nodes between both different *fnr*$^-$ data sets was considerably larger when using the ratios as input data for JProGO, which represents mainly biological effects measured by the microarray experiments (see above).

For the *lrp*$^-$ data set of Hung *et al.* (2002) the small number of common nodes for both data types comprised mainly general and specific transport processes such as '*transport*', '*porter*', '*electrochemical potential-driven transporter activity*' and '*sugar porter activity*', which all fit with the biological expectation (see chapter 3.2.2.3). Additional meaningful GO nodes identified only when using the expression ratios as input data were involved in the metabolism of specific amino acids ('*amino acid and derivative metabolism*') such as '*histidine family amino acid metabolism*' and '*histidine biosynthesis*'. In contrast to the t-test applied to the ppde, processes that represent the '*leucine biosynthesis*' were not found (see chapter 3.2.2.3). Likewise, several nodes were identified that were not directly linked to the investigated conditions, for example: '*pentose metabolism*' and '*protein folding*'.

For the two expression ratio data sets that investigated a change from aerobic to anaerobic growth conditions (Salmon *et al.*, 2003; Kang *et al.*, 2005) the overlaps with the


arcA⁻ Salmon 2005
lrp⁻ Hung 2002
fnr⁻ Salmon 2003
Wt_anaerob Salmon 2003
Wt / fnr⁻, O2 Kang 2005
Wt / fnr⁻, N2 Kang 2005
O2 / N2, Wt Kang 2005
O2 / N2, fnr⁻ Kang 2005
4 iron, 4 sulfur cluster binding
ATPase activity, coupled
ATPase activity, coupled to movement of substances
ATPase activity, coupled to transmembrane movement of substances
NAD binding
NADH dehydrogenase (quinone) activity
NADH dehydrogenase activity
RNA binding
acetyl-CoA catabolism
acetyl-CoA metabolism
aerobic respiration
alcohol catabolism
all
amine biosynthesis
amine metabolism
amine transport
amine transporter activity
amino acid and derivative metabolism
amino acid biosynthesis
amino acid derivative biosynthesis
amino acid derivative metabolism
amino acid metabolism
amino acid transport
amino acid transporter activity
amino acid-polyamine transporter activity
anion transport
anion transporter activity
aromatic amino acid family metabolism
aspartate family amino acid biosynthesis
aspartate family amino acid metabolism
behavior
binding
biogenic amine biosynthesis
biogenic amine metabolism
biological_process
biosynthesis
carbohydrate catabolism
carbohydrate transporter activity
carboxylic acid metabolism
carboxylic acid transport
carboxylic acid transporter activity
catabolism
carrier activity
catalytic activity
cation transport
cell
cell homeostasis


arcA⁻ Salmon 2005
lrp⁻ Hung 2002
fnr⁻ Salmon 2003
Wt_anaerob Salmon 2003
Wt / fnr⁻, O2 Kang 2005
Wt / fnr⁻, N2 Kang 2005
O2 / N2, Wt Kang 2005
O2 / N2, fnr⁻ Kang 2005
cell motility
cell projection
cell projection biogenesis
cell projection organization and biogenesis
cellular biosynthesis
cellular carbohydrate catabolism
cellular catabolism
cellular macromolecule catabolism
cellular macromolecule metabolism
cellular metabolism
cellular physiological process
cellular process
cellular protein metabolism
cellular respiration
cellular_component
chemotaxis
ciliary or flagellar motility
coenzyme binding
coenzyme catabolism
coenzyme metabolism
cofactor binding
cofactor catabolism
cofactor metabolism
copper ion binding
cytoplasm
disaccharide metabolism
electrochemical potential-driven transporter activity
energy derivation by oxidation of organic compounds
energy reserve metabolism
establishment of localization
fatty acid biosynthesis
ferredoxin hydrogenase activity
flagellar basal body (sensu Bacteria)
flagellum
flagellum (sensu Bacteria)
flagellum biogenesis
flagellum organization and biogenesis
glucose catabolism
glucose metabolism
glycolysis
hexose catabolism
hexose metabolism
histidine biosynthesis
histidine family amino acid biosynthesis
histidine family amino acid metabolism
histidine metabolism


arcA⁻ Salmon 2005
lrp⁻ Hung 2002
fnr⁻ Salmon 2003
Wt_anaerob Salmon 2003
Wt / fnr⁻, O2 Kang 2005
Wt / fnr⁻, N2 Kang 2005
O2 / N2, Wt Kang 2005
O2 / N2, fnr⁻ Kang 2005
hydrogen ion transporter activity
hydrogen transport
hydrolase activity, acting on acid anhydrides, catalyzing transmembrane movement of substances
inner membrane
inorganic anion transport
integral to membrane
intracellular
intracellular non-membrane-bound organelle
intracellular organelle
intrinsic to membrane
ion transport
ion transporter activity
iron ion transport
iron-sulfur cluster binding
large ribosomal subunit
localization
localization of cell
locomotion
locomotory behavior
macromolecule biosynthesis
macromolecule catabolism
macromolecule metabolism
main pathways of carbohydrate metabolism
membrane
metabolism
metal cluster binding
metal ion transport
metal ion transporter activity
molecular_function
monosaccharide catabolism
monovalent inorganic cation transport
monovalent inorganic cation transporter activity
motor activity
nickel ion binding
nickel ion transporter activity
nickel-transporting ATPase activity
nitrogen compound biosynthesis
nitrogen compound metabolism
nitrogen fixation
non-membrane-bound organelle
nucleic acid binding
organelle
organic acid metabolism
organic acid transport
organic acid transporter activity
oxidative phosphorylation

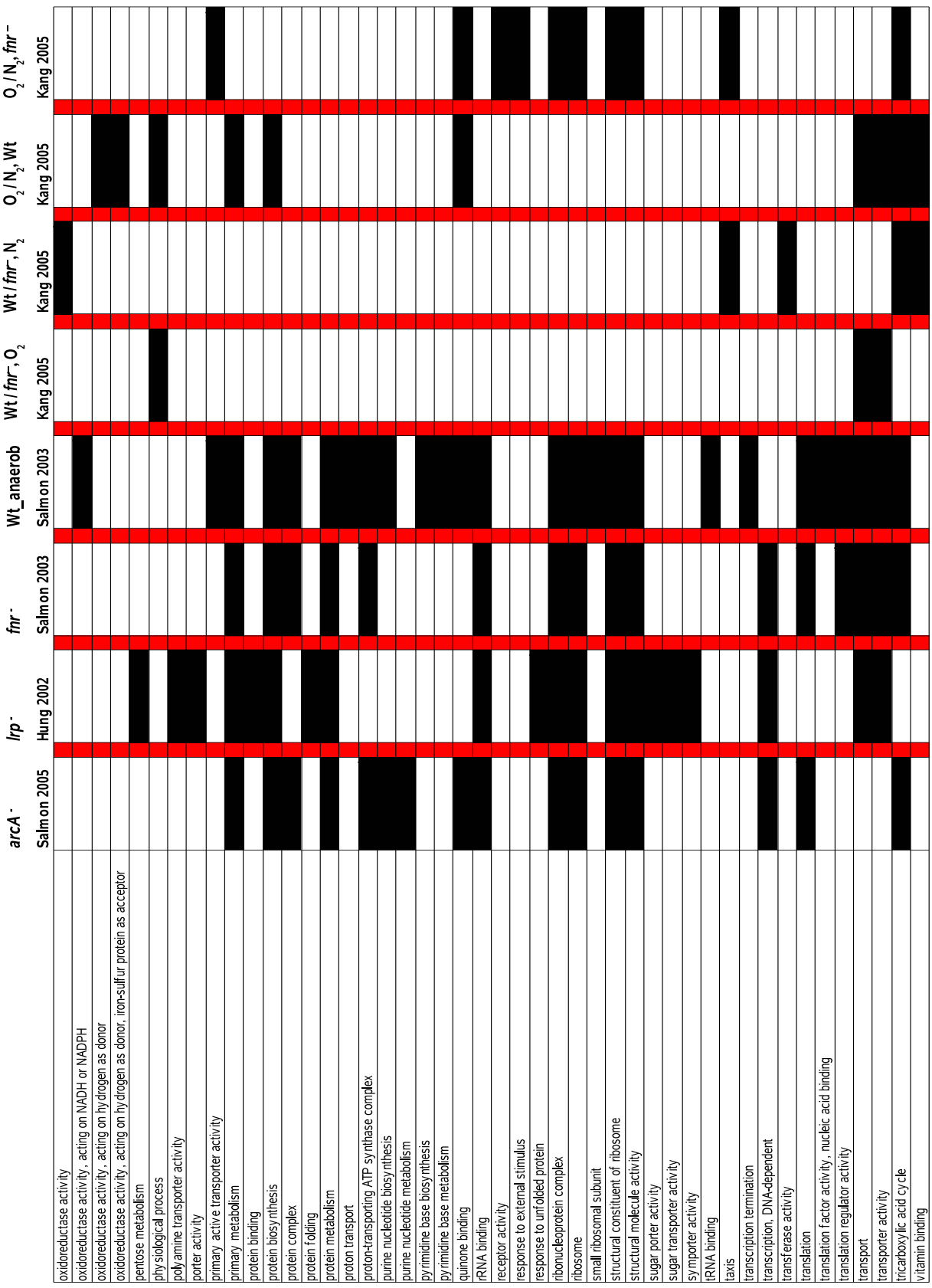


Figure 22: JProGO case study with preprocessed expression data (ratios) of *E. coli*: Significant GO nodes obtained by the Mann-Whitney U-Test (black rectangles) for the 8 tested conditions (see Tab. 6 for details). The expression ratios were provided by the authors of the studies.

results of the respective pde data were, as for the two *fnr*$^-$ data sets, uneven in size. The common significant nodes for the data of Salmon *et al.* (2003) were numerous and for those of Kang *et al.* (2005) this number was significantly smaller. In general, the common nodes covered main adaptations of the metabolism to the anaerobic growth conditions such as '*cellular respiration*' as well as '*aerobic respiration*', '*protein biosynthesis*' and '*acetyl-CoA metabolism*'. The marking of the nodes '*amino acid and derivative metabolism*' and '*amino acid transport*' as statistically significant in this context might be caused by non-biological effects such as experimental noise.

Finally, for the data set 'wild type versus *fnr*$^-$ strain, aerobic growth conditions', which served as a kind of null control (see chapter 3.2.2.3), the ratios, in contrast to the pde, marked some GO nodes as statistically significant. Since for this data set no gross changes in the transcription profile were expected (see chapter 3.2.2.3), these nodes may be explained by the ratios' higher susceptibility to noise when compared to the pde. Fortunately, predominantly GO nodes that represent rather general functions and processes such as '*biological process*', '*biosynthesis*', '*metabolism*', '*molecular function*' and '*cellular component*' were found.

Altogether, using the expression ratios, generally, more statistically significant GO nodes were obtained than with the test statistics. The additional nodes comprise both, nodes that are likely due to biological effects because they fit the biological expectation for the experiment and nodes that might be ascribed to non-biological causes such as experimental noise. Therefore, the former will contain a larger fraction of true positives hits and the latter a larger fraction of false positives. Using this nomenclature also false negatives exist. These are biologically significant GO nodes that were not detected at all, neither using ratios nor test statistics as input data.

Furthermore, the overlap of the results obtained using the ratios and test statistics was, as expected, not extensive, but after all, 25% to 75% of the nodes that were marked as significant when using test statistics were found using the ratios as input data, too. Since the common nodes constitute good candidates of affected biological functions and processes, it is recommended, when both types of expression data are available, to perform a functional interpretation (e.g. using JProGO) with both, test statistics and expression ratios. Another alternative is filtering genes by their pde and, subsequently, use only the resulting subset of genes and their expression ratios as input data. A disadvantage of this approach is the introduction of an auxiliary threshold value and the associated loss of valuable information. Thus, developing methods that integrate both sources of information, expression ratios and pde expression, like the volcano plot in the preprocessing of microarray data does, would be a worthwhile challenge.

3.2.2.5 Threshold-dependent Versus Threshold-independent Analysis

In the previous chapters the influence of the usage of different threshold-independent methods (chapter 3.2.2.2) and the impact of the type of expression data (chapter 3.2.2.4) on the outcome of the functional interpretation of expression data were examined. Another interesting question in this context is, to what extent results obtained with algorithms of the first generation, the threshold-based tests, correlate with those of the second generation, the cut-off free methods. As with the comparison of the three threshold-free tests, the Pearson's and Spearman's rank correlation was computed for the same 8 expression data sets of *E. coli* (Tab. 6). Again, the correlation coefficients were determined

Table 11: Pairwise Pearson's (r), Spearman's rank (R) correlation coefficients and Jaccard indices (JI) of the p-values of the GO nodes (p-value < 1) computed with the threshold-based Fisher's exact test and the threshold-independent unpaired Wilcoxon's test. The data sets analyzed refer to the JProGO case study in *E. coli* K-12 shown in Tab. 6 (same order). Abbreviations used are F (Fisher's exact test), U (unpaired Wilcoxon test), #F and #U (number of significant nodes of Fisher's exact and U-test) and JI (Jaccard index).

Conditions & Reference	r(F,U)	R(F,U)	#F	#U	# JI(F,U)
arcA$^-$ vs. Wt (anaerob) Salmon et al., 2005	0.683	0.721	9	10	0.727
fnr$^-$ vs. Wt (anaerob) Salmon et al., 2003	0.737	0.788	8	17	0.471
fnr$^-$ vs. Wt (anaerob) Kang et al., 2005	0.603	0.693	30	41	0.732
fnr$^-$, aerob vs.*fnr*$^-$, anaerob Kang et al., 2005	0.544	0.639	29	32	0.694
fnr$^-$ vs. Wt (aerob) Kang et al., 2005	0.259	0.267	0	0	–
lrp$^-$ vs. Wt Hung et al., 2002	0.188	0.242	4	19	0.000
Wt, aerob vs. Wt, anaerob Salmon et al., 2003	0.662	0.731	17	40	0.425
Wt, aerob vs. Wt, anaerob Kang et al., 2005	0.526	0.597	23	28	0.645
Mean	0.525	0.585			0.528
Median	0.571	0.666			0.645
Standard Deviation	0.200	0.212			0.263

for the analyzed GO nodes based on the obtained p-values. But for a fair comparison only those nodes were considered that had a meaningful p-value – regarded only as less than 1 but not equal to 1 – in the threshold-based test, which was Fisher's exact test. As representative cut-off free test the U-test was chosen, since it constitutes a good alternative to the sometimes quite stringent KS- and the often very liberal Student's t-test (chapter 3.2.2.2). Both statistical tests were performed with a two-sided alternative hypothesis. As type of expression data the test statistics (ppde and p-values) were taken, since they provide a statistically more robust estimation of the gene expression levels than expression ratios. For Fisher's exact test a threshold value of 0.9 was chosen analogously to the common error rate of 10% in statistical testing [1].

The results are shown in Table 11. With less than 0.60 the mean correlation – according to both Pearson (r) and Spearman – between Fisher's exact test and the unpaired Wilcoxon's test (Tab. 11) was clearly lower than each pairwise mean correlation between two of the three cut-off free tests ($r > 0.75$, see Tab. 8). Furthermore, the correlation bears a comparatively wide range from 0.188 to 0.737 (r) and from 0.242 to 0.788 (R), respectively. This stands in contrast with the pairwise correlation of the threshold-free tests, which covers a narrower range from about 0.65 to 0.97 (see Tab. 8). The larger de-

[1] in case of p-values, the test statistics were converted to $1 - p$values before

viation in the comparison of Fisher's exact with the U-test is also reflected by the higher standard deviation of 0.200 (see Tab. 11), which is at most only 0.113 for the correlation coefficients of the threshold-free tests. Interestingly, the threshold-based Fisher's exact test and the cut-off free U-test show a similar rank order of the GO nodes' p-values for a subset of the data sets (see rows 1,2 and 7 in Tab. 11 which have a rank correlation of > 0.7). This partial high correlation is in accordance with the considerable high overlap in the number of significant nodes computed with Fisher's exact test and the U-test for some of the expression data sets. For example, for the knockout data of the *arcA* and of the *fnr* gene, Jaccard indices (definition see chapter 3.2.2.2) of more than 0.7 were obtained, which indicate a large intersection (Tab. 11). A general trend emerges, when considering the significant nodes of both methods. The U-test identifies more significant nodes than Fisher's exact test. Furthermore, the U-test normally finds most of the nodes that have been marked as significant by Fisher's exact test.

Altogether, the results suggest that threshold-based and threshold-free methods show a weak to moderate correlation with respect to the computed p-values over all analyzed GO nodes. This fits with the expectation since both represent different types of statistical tests, either determining the over-representation of preselected genes (Fisher's exact test) or the difference between two empirical probability distributions (U-test). Remarkably, when focusing only on the significant nodes, using the same error rate and method of correction for the multiple testing effect, the overlap between the cut-off based and the cut-off free method was often large for the selected cut-off value of 0.9. However, the cut-off based method in each case misses to identify significant nodes that are found by the cut-off free method. This observation and the described problem of the influence of the threshold value on the outcome of the analysis (chapter 3.2.1) rather argue against the use of cut-off based methods like Fisher's exact test in favor of threshold-free methods.

3.2.3 Successful Employment of JProGO for a Time Series Study on *B. subtilis* Spore Germination and Outgrowth

In the following the application of JProGO to expression data from the Gram-positive prokaryotic model organism *B. subtilis* (strain 168), is discussed. For this purpose, the results of an analysis that was performed with the help of the JProGO web server by others (Keijser *et al.*, 2007) were taken. The corresponding publication was found through a search with the Pubmed and Scopus database. The authors describe an in-depth time series analysis of *B. subtilis* spore germination and outgrowth (Keijser *et al.*, 2007). Expression profiling experiments were measured at several sporulation and outgrowth time points using vegetative cells as a common reference. For the functional interpretation with JProGO, the authors employed the default settings comprising an FDR of 0.05 and a two-sided U-test, analogously to the case study performed with *E. coli* (see chapter 3.2.2.2). The results of Keijser *et al.* (2007), which include a selection of the significant GO nodes (see Fig. 23, adapted from Keijser *et al.* (2007)), are discussed in the following.

During the early phase of outgrowth (5 to 30 min.) the identified GO groups include transport functions (e.g. '*transporter activity*'), '*regulation of transcription*', '*DNA replication*', '*DNA repair*', '*RNA modification*', '*heterocycle biosynthesis*' and the process of '*sporulation*' itself (Fig. 23). The alterations in the *transcriptional regulation* correspond to the observed induction of many genes that are involved in transcription and its regulation like the RNA polymerase sigma factors (SigY, SigI) and the anti-terminator NusG (Keijser *et al.*, 2007). The *transport processes* contained the genes for several multi-drug

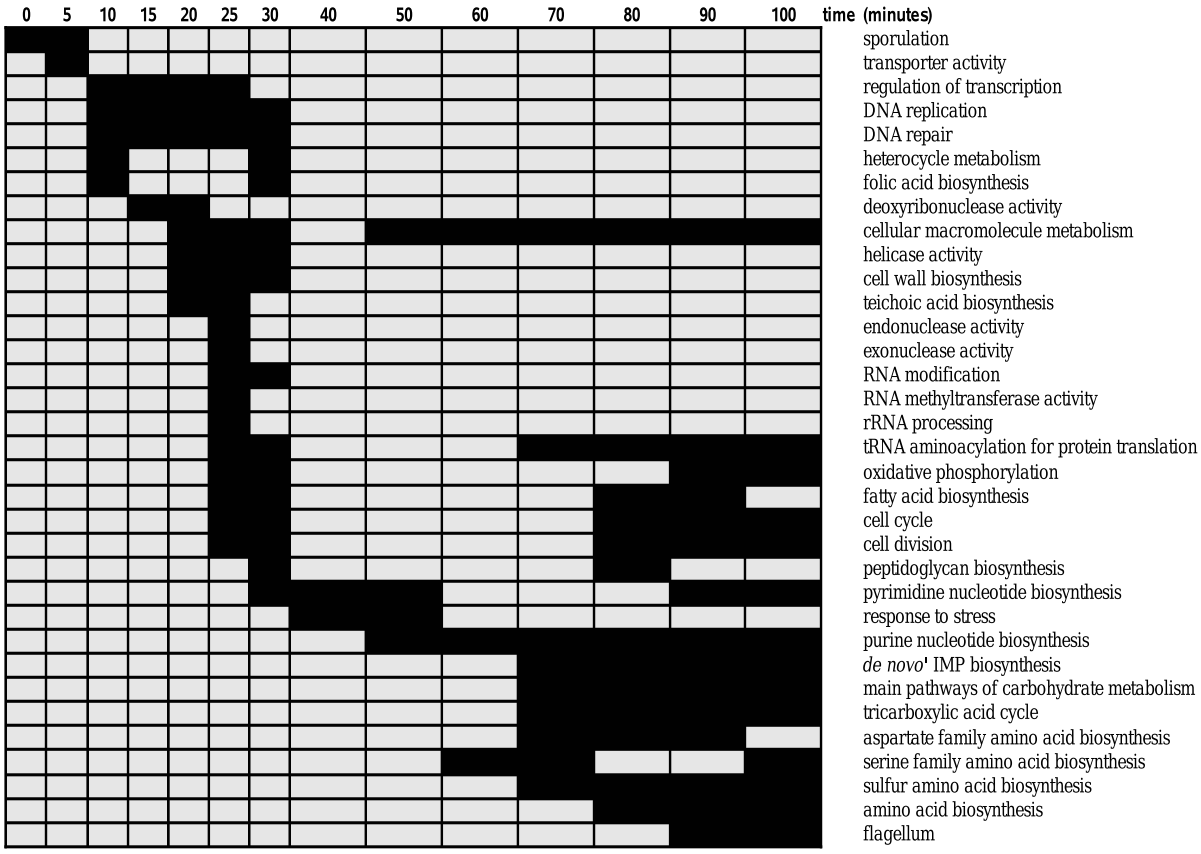


Figure 23: Results of a JProGO analysis for the expression data from a time series of *B. subtilis* spore germination and outgrowth performed by Keijser *et al.* (2007). Statistically significant GO nodes are marked by filled black rectangles. Expression data from vegetative cells served as a common reference for all sporulation time points (0 – 100 min.). The figure was adopted from Keijser *et al.* (2007).

and ABC transporters as well as Na^+/H^+ antiporters, which presumably ensure a rapid supply of the germinating spore with substances needed for efficient outgrowth, e.g. amino acids, sugars and other essential metabolites (Keijser *et al.*, 2007). In addition, since the H^+ transporter genes *ktrA*, *ktrD* and the glycine betaine transporter genes *opuA*/*opuB* were found up-regulated (Keijser *et al.*, 2007), a defense against osmotic stress (Holtmann *et al.*, 2003) might explain the statistical significance of these GO nodes. The found *DNA repair and replication* functions might be due to the observed expression of nucleotide excision repair enzymes (e.g. *uvrA*, *uvrB*), base repair and helicase/exonuclease enzymes. Some of these have been shown to protect the germinating spores against harmful influences such as UV radiation and heat (Nicholson *et al.*, 2000).

In the second intermediate phase of outgrowth (25 to 50 min.), during which about 400 genes were up-regulated, the statistically significant GO nodes comprised '*DNA replication*', '*DNA repair*', '*cell cycle*', '*cell division*', *fatty acid biosynthesis*', '*peptidoglycan biosynthesis*' and '*response to stress*' (Fig. 23). The found *cell growth and division* functions could be explained by the fact that the outgrowing spore cells prepare for the cylindrical growth, chromosomal segregation and cell division (Keijser *et al.*, 2007). Consistently, corresponding genes and operons, such as *mreBHCD* and *minCD*, were induced during this period, in addition to genes involved in the *biosynthesis of fatty acids*. Furthermore, genes that mediate *chromosome condensation and segregation* were activated in this time frame (Keijser *et al.*, 2007). In addition, a second phase of *DNA repair* took place, and the respective genes, e.g. *exoA*, *splAB* and *ung*, were expressed. The observed activation of the more general GO group *stress response* after 40 to 50 min. was due to an up-regulation of genes that belong to the sigma B regulon, whereas the initial trigger for this response is not obvious and might be revealed in subsequent studies (Keijser *et al.*, 2007).

In the late phases of outgrowth (50 to 80 min. and 80 to 100 min.) the relevant GO nodes include those that represented the biosynthesis of nucleotides ('*pyrimidine nucleotide biosynthesis*', '*pyrimidine nucleotide biosynthesis*') and, later the '*peptidoglycan biosynthesis*'. The activation of genes involved in the biosynthesis of purines and pyrmidines, observed by the authors, had a peak around 50 and 90 min. In addition, especially in the second of the late phases (80 to 100 min), GO nodes were found that represent typical functions of an actively growing vegetative cell such as , '*flagellum*', groups reflecting amino acid metabolism, e.g. '*sulfur amino acid biosynthesis*', '*serine family amino acid biosynthesis*' and '*aspartate family amino acid biosynthesis*' as well as '*cell division*', '*oxidative phosphorylation*', '*glycolysis*' and '*tricarboxylic acid cycle*' (Fig. 23). These nodes also fit the expectation of Keijser *et al.* (2007).

Altogether, the study of genome wide expression profiling in *B. subtilis* (Keijser *et al.*, 2007) verifies the potential of JProGO in providing a fast overview on the transcriptionally altered cellular processes and functions. Furthermore, it demonstrated the suitability of the tool for more complex experimental set-ups than simple two-conditions comparisons like time series analyses and for other prokaryotic organisms than *E. coli* that are also well annotated.

3.3 Combined Low-, Mid- and High-Level Analysis of Prokaryotic Microarray Raw Expression Data Using Bioconductor and JProGO

After the in-depth evaluation of the JProGO program and its algorithms for the functional interpretation of preprocessed, previously published prokaryotic microarray data (see previous chapter), in the following, JProGO was used in a combined low-, mid- and high-level analysis of raw expression data from in-house expression profiling experiments. This analysis includes the preprocessing of the raw data using different normalization and background subtraction algorithms, the computation of the probabilities of differential expression and the subsequent functional interpretation using the JProGO tool. In this context, the influence of different preprocessing algorithms and expression data types on the outcome of the functional interpretation was investigated.

3.3.1 Low-Level Analysis: Preprocessing of the Raw Expression Data Using Different Algorithms

The microarray experiments, which were all preprocessed using the Bioconductor package (chapter 2.3.2), comprised both principal types of expression profiling experiments: firstly, alterations of cultivation conditions and, secondly, gene knockouts as well as combinations of both (Tab. 12). The wild type strain grown under otherwise identical conditions served as reference for comparison. The used microarray platform was Affymetrix GeneChip®, the analyzed organism *P. aeruginosa* (PAO1) and, in each case, three biological replicates were performed by the experimenters (Tab. 12).

Table 12: Overview on the in-house microarray expression raw data sets that were analyzed. All experiments were conducted (see Experimenter column) with the bacterium *P. aeruginosa* (PAO1) and for each experimental condition three replicates measurements were performed. The data were preprocessed with Bioconductor, first, and then analyzed with JProGO. Wt designates the wild type strain.

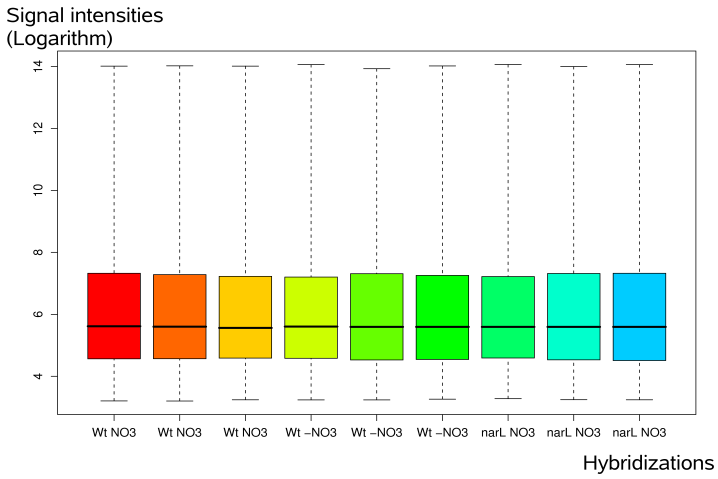
Experiment & Conditions	Experimenters & Reference
Wt, aerob, log phase	Schreiber *et al.*, 2006, unpubl. results
Wt, anaerob, 1d pyruvate fermentation	Schreiber *et al.*, 2006, unpubl. results
Wt, anaerob, 7d pyruvate fermentation	Schreiber *et al.*, 2006, unpubl. results
uspK$^-$, anaerob, 1d pyruvate fermentation	Schreiber *et al.*, 2006, unpubl. results
Wt, anaerob, with nitrate	Benkert *et al.*, 2008, unpubl. results
Wt, anaerob, without nitrate	Benkert *et al.*, 2008, unpubl. results
narL$^-$,, anaerob, with nitrate	Benkert *et al.*, 2008, unpubl. results
Wt, biofilm, artificial sputum medium	Thoma *et al.*, 2007, unpubl. results
Wt, biofilm, LB medium	Thoma *et al.*, 2007, unpubl. results
Wt, anaerob, log phase	Bös *et al.*, 2007, unpubl. results
relA$^-$/*narL*$^-$, anaerob, log phase	Bös *et al.*, 2007, unpubl. results
Wt, alkali stress	Bös *et al.*, 2007, unpubl. results
relA$^-$/*narL*$^-$, alkali stress	Bös *et al.*, 2007, unpubl. results

A descriptive statistical inspection using the affy package of Bioconductor (Gautier *et al.*, 2004) revealed that on average around 30% of the probe sets of the 12 arrays of the unpublished results of Schreiber *et al. (2006)* (Tab. 12) had a mismatch signal that was higher than that of the respective perfect match (see chapter 1.2.2). This fraction was concordant with the 31% to 35% reported for a large collection of GeneChip® data of the eukaryotic model organisms fruit fly, mouse and man (Naef *et al.*, 2001, 2002). The above mentioned finding and the open question, what is actually measured by the mismatch signals (Naef *et al.*, 2001, 2002; Irizarry *et al.*, 2003b), argued against the application of algorithms such as Affymetrix MAS4 and MAS5 (Affymetrix Inc., 2001, 2002), which use the mismatches for computing the expression values (Bolstad *et al.*, 2005b). Therefore, for the subsequent analyses more recently developed preprocessing algorithms were selected that solely make use of the perfect match probe sets and ignore the mismatch probe sets. These are the rma (Irizarry *et al.*, 2003a,b), the vsn (Huber *et al.*, 2002) and the dChip (Li and Wong, 2001a,b) method. This is in agreement with several recent publications that abandon the mismatch-based chip-by-chip algorithms MAS4/MAS5, that are e.g. provided by the Affymetrix software, and prefer instead perfect match-based, model fitting methods (see also Bolstad *et al.*, 2005b; Wu and Irizarry, 2005; Seo and Hoffman, 2006).

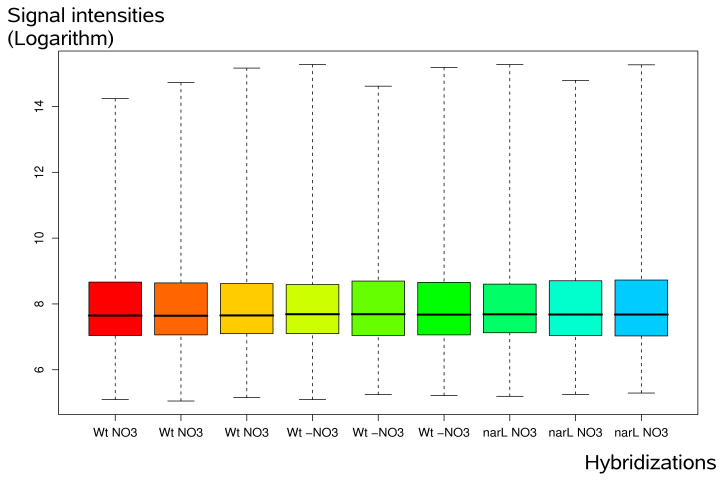
Since it was not clear, which of the perfect match-based preprocessing algorithms was best suited for the raw data, and due to the general problem of finding the preprocessing method best suited for a given microarray experiment (see e.g. Millenaar *et al.*, 2006; Seo and Hoffman, 2006), several of the commonly used methods (Seo and Hoffman, 2006) were tested. For this purpose, a subset of the microarray data sets was taken (data of Benkert et al., (unpubl. res.), 2008, see below). The selected methods comprised:

- the rma method introduced by Irizarry *et al.* (2003a,b)
- the vsn method of Huber *et al.* (2002)
- the dChip method developed by Li and Wong (2001a,b)

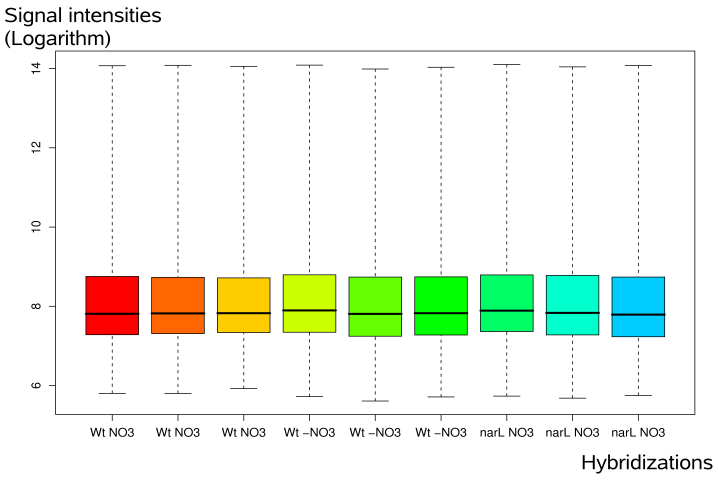
In the following, the results of the preprocessing of the data sets of Benkert et al., 2008 (unpublished results, see Tab. 12) were described in detail. One reason for selecting these data sets was that their experimental conditions comprise a knockout of the nitrate reponsive transcriptional regulator (NarL) and that cultures were challenged with drastic changes in cultivation conditions (aerobic to anaerobic growth). Therefore, significant pertubations of the gene expression profiles were expected, what would make the preprocessed data especially suited (see chapter 3.2.2) for the subsequent functional interpretation with JProGO.



a)



b)



c)

Figure 24: Box plots showing the signal intensities of the preprocessed data of Benkert et al., 2008 (unpubl. res.). As preprocessing methods a) rma (Irizarry *et al.*, 2003b) b) vsn (Huber *et al.*, 2002) and c) dChip (Li and Wong, 2001a) were used. All plots refer to logarithmic expression levels. The three replicates of *P. aeruginosa* (PAO1) wild type cells grown anaerobically with nitrate (Wt NO_3^-), without nitrate (Wt $-NO_3^-$) and the *narL* mutant grown with nitrate (narL NO_3^-) are shown. The upper and lower border of the colored boxes denote the upper and lower quartil of all measured values and the horizontal bar in each box the corresponding median. The maximum and minimum value is in each case represented by the upper and lower bounding line.

Quality control I: Inspection of the raw data and expression value distribution of the preprocessed data

At first, several visual inspections and statistical computations were performed to assess the quality of the raw data and the preprocessing steps. They include the checking of the raw image data and the creation of diagnostic plots like RNA degradation (Fig. 30, Appendix), box (Fig. 24) and density plots (Fig. 31, Appendix). A look at the raw images of the arrays containing either the original or logarithmically transformed data did not reveal any severe spatial artifacts. The next step was the inspection of the RNA degradation plots: The obtained lines revealed for the average probeset intensities of each of the 9 raw data sets a slight overall degradation with a slope of around -1, which proceeds from the 3' to 5' (Fig. 30, Appendix). This fits the expectation since the RNA polymerase starts with the synthesis of the 5' end of the mRNA and degradation begins at the 3' end (Steege, 2000; Deutscher, 2006). In contrast, RNA degradation plots obtained for microarrays of man and other eukaryotes, which were also based on the Affymetrix platform, showed a degradation from the 5' to 3' end (Bolstad *et al.*, 2005a). This is in agreement with the presence of 5' - 3' exoribonucleases in eukaryotes, which do not play such a major role in bacteria and archaea (Steege, 2000; Deutscher, 2006). Compared to the sign and actual slope of the RNA degradation graphs the agreement between the different arrays (Bolstad *et al.*, 2005a) is more important. Since the lines of all 9 arrays had a similar shape and slope (Fig. 30, Appendix) of around -1 (computation with R, data not shown), comparisons of genes across arrays should still be valid after preprocessing the raw data (Bolstad *et al.*, 2005a). Besides the arrays of Benkert et al., 2008 (unpublished results), a similar slope and shape was observed for the other raw data sets analyzed (Tab. 12).

After the raw data, the preprocessed data were inspected using box- and density plots. All three preprocessing methods – rma, vsn and dChip – yielded unimodal distributions of the log-transformed intensity values (Fig. 31, Appendix) with a peak near the median value (Fig. 24 and 31, Appendix). Compared to the density function of a Gaussian distribution, the peaks were shifted to the left giving rise to asymmetric distributions. Such deviations from normal-like distributions of the gene expression levels are commonly described in the literature (see Xiao *et al.*, 2006) including unimodal asymmetric intensity histograms (Bolstad *et al.*, 2005a) which were also observed in this study. All three preprocessing methods spanned a similar range of intensity values from around 5.5 to 14, whereas the rma method had a slightly broader range containing already intensity values of less than 4 (Fig. 24). The latter method also showed a broader interquartile range – the middle 50 % of the data – which was shifted to slightly lower values when compared to the other two methods and its median was smaller, too. Maybe the lack of a background correction step in the other two methods – vsn and dChip – could be the reason for the observed differences. Altogether, the density and box plots as well as the visual inspection of the raw chip images indicated a good quality of the analyzed data sets and a reliable preprocessing procedure.

Quality control II: Agreement of the replicates

The next steps served as further quality control whereas their focus was on the elucidation of the correlation and reproducibility between the replicate arrays. For this purpose, for each preprocessing method the pairwise scatter plots of all 9 data sets were generated and the corresponding pairwise correlation coefficients over all genes were computed (not shown). For all three preprocessing variants, the correlation coefficients were highest

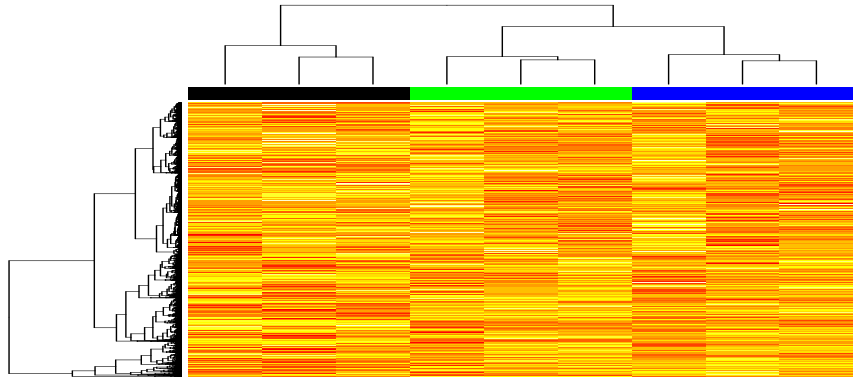
between the replicate data sets, respectively. For the rma method the correlation was at least 0.99, for the dChip algorithm 0.98 and for the vsn method at least 0.98. For the 3 non-replicate pairwise comparisons the correlation was in each case lower. Interestingly, the weakest overall correlation, 0.94 - 0.96 for the three methods, was between wild type cells grown anaerobically with and those grown without nitrate. The comparison of these two wild type cultivation conditions with the *narL* mutant grown anaerobically with nitrate yielded a slightly higher correlation between 0.96 and 0.98. The smaller similarity between both wild type cell grown with and without NO_3^- could be caused, amongst the different NarL activity, by the action of additional transcriptional regulators like Dnr. These factors do not change their activity when comparing the wild type and the *narL* knockout strain, both grown under anaerobic conditions, and therefore the correlation is higher in this case.

The scatter plots confirmed the good reproducibility between the replicate array experiments that was found by the correlation analysis. Exemplarily, for the rma method one preprocessed data set – the wild type strain grown anaerobically with nitrate (Wt $+NO_3^-$) is selected and its comparison with all 8 other arrays is shown (see Fig. 32, Appendix): Many genes with a log-ratio of more than 2 or less than -2 (green lines in Fig. 32, Appendix) were found, when Wt $+NO_3^-$ is compared to Wt $-NO_3^-$ (Wt_minus_NO3 in Fig. 32, Appendix) and the *narL* mutant grown with nitrate (NarL_NO3 in Fig. 32). In contrast, this proportion was much lower for the two replicate arrays of Wt $+NO_3^-$ (Wt_NO3 in Fig. 32). Analogous results were obtained for the other data sets and preprocessing methods.

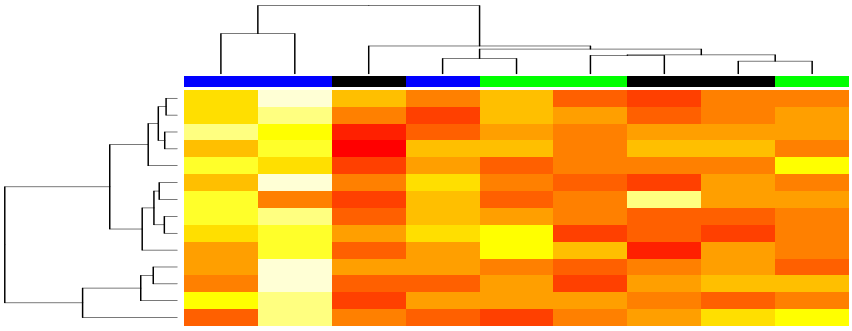
Finally, as the last quality control to investigate the agreement between the replicated conditions, the array data were clustered hierarchically and the results were visualized as heatmaps (Fig. 25, see also chapter 2.8 for background information). The heatmap representation, which was introduced by Eisen *et al.* (1998) for a visualization of microarray data (see also Huber *et al.*, 2005), allows the simultaneous representation of the clusters obtained for the experimental conditions and for the genes. A heatmap was computed for each of the following three sets of genes:

- all 5900 probe sets spotted on the PAO1 Affymetrix GeneChip®
- a subset of 14 negative control genes derived from other organisms
- a subset of 14 randomly selected genes

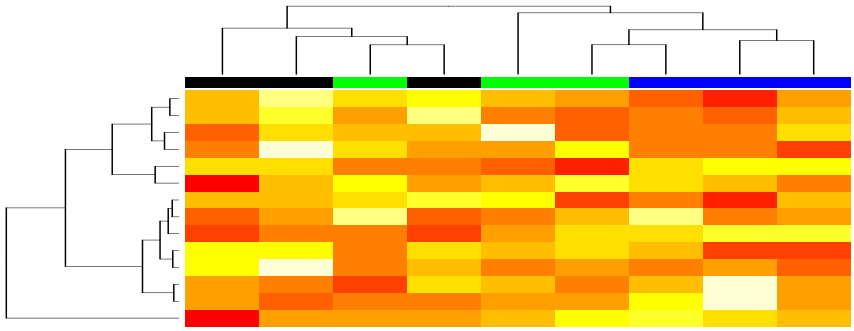
The results for the rma method are depicted in Fig. 25: When using all 5900 probe sets, the three replicate arrays of each condition clustered together, thus a clear separation between the three experimental conditions was obtained (Fig. 25a). When comparing the different conditions, interestingly, the wild type strain grown without nitrate and the *narL* mutant grown with nitrate showed the highest agreement, which is consistent with the results from the correlation analysis (see above). The heatmap for the 14 negative control genes (Fig. 25b) completely destroyed the clustering of the three different growth conditions observed above. The heatmap for the 14 randomly selected genes, in turn, partially reconstituted these clusters (Fig. 25c) that were obtained when analyzing all 5900 probe sets. This indicated that random noise, which is inherent in microarray raw data, was successfully filtered by the preprocessing steps. Altogether, the scatter plots, the correlation analysis and the heatmaps verified the good agreement between the replicate arrays after preprocessing with the rma method and indicated that the observed differences between the three experimental conditions were mainly due to differences in



a)



b)



c)

Figure 25: Heatmaps of the preprocessed data of Benkert et al., 2008 (unpubl. res.). The dendrograms from the clustering of the experimental conditions (x-axis or columns) and of the genes (y-axis or rows) are shown. The following sets of genes were used for clustering: a) all 5900 probe sets of the Affymetrix *P. aeruginosa* (PAO1) GeneChip®, b) a subset of all genes (a) containing 14 negative control genes derived from other organisms and c) a subset of all genes (a) containing 14 randomly selected genes. The colored bars represent three anaerobic growth conditions: black is the wild type strain grown with NO_3^-, green the wild type strain grown without NO_3^- and blue the *narL* mutant grown with NO_3^-.

Table 13: Correlation between the expression ratios of the data of Benkert et al., 2008 (unpubl. res.) obtained by the three different preprocessing methods (see Tab. 14). For each pairwise comparison of experimental conditions the Pearson's correlation coefficient was computed for all 5900 probe sets.

Compared conditions	rma vs. dChip	rma vs. vsn	vsn vs. dChip
Wt $+NO_3^-$ vs. Wt $-NO_3^-$	0.92	0.98	0.93
narL$^-$ $+NO_3^-$ vs. Wt $-NO_3^-$	0.95	0.96	0.97
narL$^-$ vs. Wt, both $+NO_3^-$	0.92	0.97	0.93

the gene expression profiles. For the other two employed preprocessing methods used, vsn and dChip, comparable results were obtained using scatter plots, correlation analysis and heatmaps (not shown for lack of space). For these methods, likewise the individual replicate conditions were most similar to each other.

Comparison of the computed expression ratios

After the successful preprocessing of the expression data of Benkert et al., 2008 (unpubl. res.) with the rma, vsn and dChip methods, the corresponding expression ratios were computed for each pairwise comparison of the three experimental conditions (partially listed in Tab. 14). In order to determine the overall similarity between the expression ratios obtained for the different preprocessing methods, the Pearson's correlation coefficients were determined. In general, the overall correlation was high for each comparison and constituted at least 0.92 (Tab. 13). The observed correlation was highest for the comparison of the rma and the vsn method, which bore a minimal correlation coefficient of 0.96. These findings fit with the expectation and are in agreement with a previous benchmark study using a huge number of reference data sets from eukaryotes (see clustering tree from Irizarry *et al.*, 2006).

For an overview on the genes that were most up- and down-regulated according to the computed expression ratios, the 10 genes belonging to the top and the bottom of the expression matrices were depicted in Table 14. Even within these small subsets the intersection of genes that were found with all three methods was comparatively high ranging from 60% to 80%. At the same time, with this restricted view, the differences between the individual preprocessing methods remained more prominent in comparison to the high overall agreement ($r > 0.91$) computed over all genes of the chip (see above). The degree of correlation of the results obtained in a JProGO-based analysis using the expression data from the different preprocessing methods was investigated below (chapter 3.3.3).

Another benefit of determining the top 10 up- and down-regulated genes (Tab. 14) was the generation of a list of potentially interesting candidate genes that can be confirmed in subsequent analyses, e.g. by determining the pde or other test statistics (see chapter 3.3.2). Since these genes represent only a small fraction of the relevant genes and a more comprehensive biological interpretation of the data can be obtained when looking at groups of functional related genes such as GO nodes (chapter 3.3.3), in the following only the top ranking genes that were found by all three methods were shortly discussed:

- Wt $-NO_3^-$ vs. Wt $+NO_3^-$: The down-regulation of the *narG*, *H* and *I* genes, which code for the subunits of the respiratory nitrate reductase, in the absence of nitrate

Table 14: Gene expression ratios obtained for the microarray data of Benkert et al., 2008 (unpubl. res.) after preprocessing with either the rma, vsn or dChip method. Due to the lack of space only the 10 most up- and down-regulated genes are shown. For each pairwise combination of conditions, the intersection of all three preprocessing methods – the genes which were present in the 10 most up- and down-regulated genes in all methods – is represented by *italicized gene names*. Genes designated with *ig* represent intergenic regions (genomic positions not shown).

Wildtype -NO_3^- /Wildtype +NO_3^-

up-regulated genes						down-regulated genes					
rma		**vsn**		**dChip**		**rma**		**vsn**		**dChip**	
gene	ratio	gene	ratio	gene	ratio	gene	ratio	gene	ratio	gene	ratio
norB	0.01	*norB*	0.02	*norB*	0.02	*PA1746*	9.30	*PA1746*	6.76	*PA1746*	5.80
norC	0.01	*norC*	0.03	*norC*	0.03	*PA3284*	6.13	5SrRNA	3.76	*PA3284*	3.86
narH	0.02	*narI*	0.05	*narG*	0.06	5SrRNA	4.76	*PA3284*	3.72	*rmf*	2.85
narI	0.02	*nosZ*	0.05	*narJ*	0.06	PA3283	4.33	*rmf*	3.4	*ig*	2.83
narG	0.02	*narH*	0.05	*narH*	0.06	*rmf*	3.90	*PA0141*	2.87	*PA0141*	2.80
narJ	0.02	*narG*	0.05	*PA1856*	0.07	*PA0128*	3.54	*cspD*	2.84	*cspD*	2.73
PA0525	0.02	*narJ*	0.06	*narI*	0.07	*PA4611*	3.42	*PA4611*	2.68	PA5460	2.68
nosZ	0.03	*PA0525*	0.07	*PA0525*	0.07	*cspD*	3.39	*PA0128*	2.56	*PA0128*	2.64
PA1856	0.03	*PA1856*	0.09	*nosZ*	0.07	*PA0141*	3.39	PA3283	2.38	PA2482	2.52
narK1	0.03	*narK1*	0.09	*narK1*	0.08	PA4377	3.17	PA5475	2.35	*PA4611*	2.51

***narL*$^-$ +NO_3^- / Wildtype -NO_3^-**

up-regulated genes						down-regulated genes					
rma		**vsn**		**dChip**		**rma**		**vsn**		**dChip**	
gene	ratio	gene	ratio	gene	ratio	gene	ratio	gene	ratio	gene	ratio
PA3284	0.09	*PA2128*	0.17	*PA3284*	0.17	*norB*	58.39	*norB*	20.01	*norB*	19.89
PA2128	0.10	*PA3284*	0.19	*PA2128*	0.21	*norC*	37.99	*nosZ*	19.72	*norC*	19.26
PA3283	0.12	*PA3283*	0.27	*PA4685*	0.26	*nosZ*	34.66	*norC*	19.09	*nosZ*	12.61
PA4685	0.14	*PA4500*	0.28	*PA4500*	0.28	*nosD*	21.54	*nosD*	7.93	*nosF*	8.35
PA4500	0.14	*PA4685*	0.30	*PA3283*	0.29	*nosF*	17.93	PA3205	7.81	*nosD*	7.24
narL	0.21	*narL*	0.37	*PA2129*	0.36	PA0525	13.57	nirM	7.14	*PA0526*	6.33
PA2129	0.28	*PA2129*	0.40	*narL*	0.37	*PA0526*	12.77	*nosF*	6.97	nosL	6.13
kdpB	0.37	5SrRNA	0.46	PA2767	0.40	nosY	10.07	*PA0526*	6.64	nirM	5.83
ig	0.38	ig	0.47	kdpB	0.41	nosR	9.68	nirS	6.49	nirC	5.62
kdpC	0.38	PA1555	0.51	kdpF	0.49	PA0513	9.40	nirC	6.33	PA3205	5.61

***narL*$^-$/Wildtype, both +NO_3^-**

up-regulated genes						down-regulated genes					
rma		**vsn**		**dChip**		**rma**		**vsn**		**dChip**	
gene	ratio	gene	ratio	gene	ratio	gene	ratio	gene	ratio	gene	ratio
narH	0.02	*narI*	0.05	*narG*	0.06	*PA1746*	6.72	*PA1746*	5.14	*PA1746*	4.45
narG	0.02	*narH*	0.06	*narH*	0.06	*PA4739*	5.38	*rmf*	4.14	*rmf*	3.21
narJ	0.02	*narG*	0.06	*narJ*	0.06	*PA3575*	5.26	*PA3205*	3.77	PA4607	3.15
narI	0.02	*narJ*	0.07	*narI*	0.08	*rmf*	4.7	*PA3691*	3.26	*PA3691*	3.11
PA3871	0.03	*narK1*	0.09	*PA1856*	0.08	fimU	4.41	*PA3575*	3.24	fptA	3.11
narK1	0.03	*PA1856*	0.10	*narK1*	0.08	*PA3205*	4.02	PA4607	3.21	PA3692	3.00
PA1856	0.04	PA3871	0.10	ig	0.09	fptA	3.93	*PA4739*	3.15	PA4880	2.89
moaB1	0.05	*moaB1*	0.13	*narK2*	0.09	PA4141	3.92	PA4141	2.73	*PA3205*	2.87
narK2	0.05	*narK2*	0.13	*moaB1*	0.12	*PA3691*	3.90	PA4880	2.70	*PA3575*	2.78
PA1854	0.06	PA1855	0.14	moaA1	0.12	PA1323	3.87	PA3819	2.69	*PA4739*	2.73

clearly correspond to the expectation. Furthermore, *narK1* and *narJ* belong to the same operon (Sharma *et al.*, 2006) and were likewise found down-regulated (see also Schreiber *et al.*, 2007).

- *narL*$^-$ +NO_3^- vs. Wt -NO_3^-: The down-regulation of the gene coding for the transcriptional regulator *narL* fits the *a priori* expectation when comparing the knockout strain with the wild type. Furthermore, the genes *norZ*, *norD*, *norF*, which all belong to the *norBCDFZ*, were jointly up-regulated (see also Schreiber *et al.*, 2007).

- *narL*$^-$ vs. Wt, both +NO_3^-: The observed strong down-regulation of the whole *narK1K2GHJI* operon in the *narL* knockout strain under anaerobic denitrifying fits with the previously described finding that this operon is transcriptionally activated by NarL in *P. aeruginosa* (Krieger, 2001; Schreiber *et al.*, 2007).

Altogether, the genes with the highest and lowest expression ratios that were obtained for the rma, vsn and dChip preprocessing method were consistent with the biological expectation with respect to the analyzed experimental conditions. In the next chapter (chapter 3.3.2), the probabilities of differential expression were computed for the data sets and compared to the expression ratios.

3.3.2 Mid-level analysis: Computation of the Probabilities of Differential Expression

The preprocessing and computation of the expression values for the data sets of Benkert et al., 2008 (unpubl. res., see chapter 3.3.1) was the basis for the determination of the pde, here the ppde, which is described in this chapter. In contrast to the expression ratios the pde are not solely based on the means of the replicate expression values, but they also consider the variances of these values (see chapter 1.4.2). Therefore, the pde are another important measure of preprocessed expression data which, in contrast to the expression ratios, reflect fluctuations in the gene expression values and yield more reliable values especially for genes with lower signal intensities.

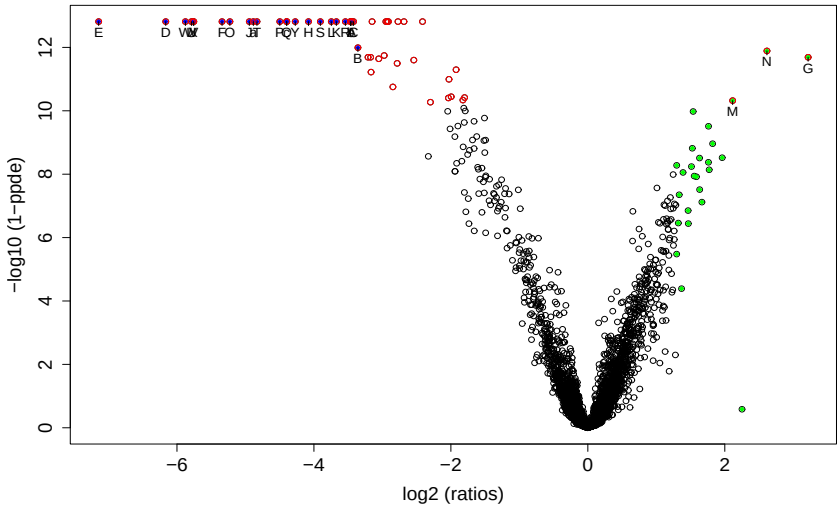
Using the 9 expression data sets of Benkert et al., 2008 (unpubl. res.) that were preprocessed with either the rma, vsn or dChip method (chapter 3.3.1), the posterior probabilities of differential expression (ppde) for all 3 comparisons of the three different experimental conditions (for an itemization see Tab. 14) were computed according to the method of Baldi and Long (2001) and Hatfield *et al.* (2003). Due to the lack of space, only the results for the data preprocessed with the rma method are shown (Tab. 15). As expected for the two comparisons that comprise the *narL* mutant strain, the *narL* gene was found among the genes that showed the strongest differential expression (Tab. 15). Analogous findings were obtained for the dChip and the vsn method. Moreover, many genes were found at both, the top of the list of genes with the best ppde (Tab. 15) and the highest or lowest expression ratios (Tab. 14), respectively: For example, when comparing wild type cells grown anaerobically with and without nitrate, genes coding for the respiratory nitrate reductase, *narGHI*, as well as *norB*, *norC* and *nosZ* had the highest ppde (Tab. 15, left column). And for the comparison of the *narL* knockout strain grown with NO_3^- and the wild type strain grown with NO_3^- , many genes of *narK1K2GHJI* operon were also found differentially expressed (Tab. 15, right column).

This correlation of genes that possess both, a high absolute log expression ratio and a high pde, was previously observed for other microarray data sets and can be visualized as

Table 15: Genes with the top 40 ppde obtained for the gene expression data of Benkert et al., 2008 (unpubl. res.) after preprocessing with the rma method. Due to lack of space the remaining genes (> 5800) are not shown. The genes are sorted in descending order by their ppde (values not shown). The minimal ppde is denoted in the last row. The columns represent the three pairwise comparisons of the analyzed conditions (see Tab. 14). Genes denoted in **bold** have the same ppde.

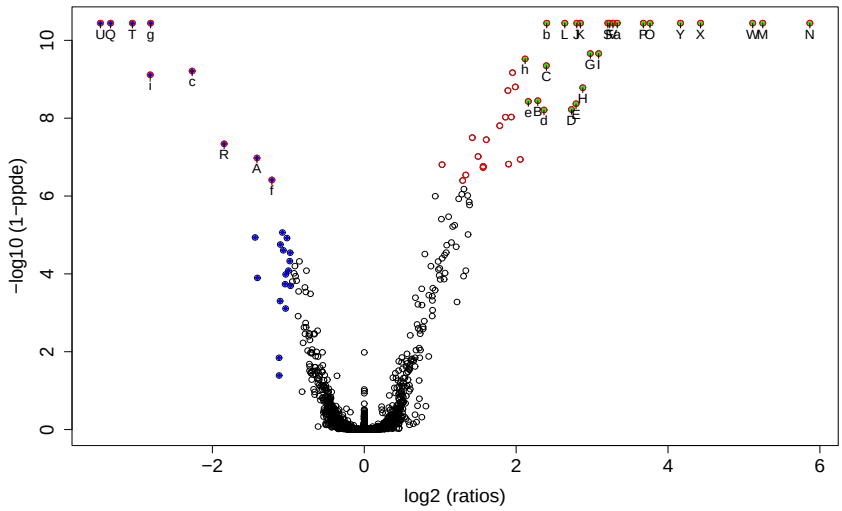
	Wt -NO_3^- vs Wt +NO_3^-	*narL*$^-$ +NO_3^- vs Wt -NO_3^-	*narL*$^-$ vs. Wt, both +NO_3^-
	PA0525	**PA0526**	**PA4685**
	norC	**PA3284**	**narK1**
	PA0521	**PA0513**	**narK2**
	nirS	**norB**	**narH**
	narK1	**PA4500**	**moaB1**
	PA1854	**nosF**	**narG**
	nosY	**PA3283**	**PA3871**
	nirC	**nosR**	**PA4500**
	PA3871	**nosL**	**moeA1**
	narG	**nirC**	**PA1854**
	moaA1	**PA3205**	**PA1855**
	norB	**nosZ**	**PA1856**
	nirM	**nosD**	**narJ**
	narI	**norC**	**moaA1**
	PA1855	**PA2128**	**narI**
	nirQ	**nirS**	**oprE**
	narK2	**nirM**	fimU
	narH	**nosY**	PA3575
	PA0513	**PA0525**	PA2128
	PA3913	nirF	putA
	moeA1	nirL	PA4739
	moaB1	fimU	PA3205
	nosF	PA0510	PA2663
	nosR	narL	PA1746
	PA2663	PA3530	rmf
	fhp	PA4685	PA4205
	PA1856	PA0521	pilW
	narJ	PA0515	PA4207
	nosL	PA3912	PA3819
	nosD	nirN	fhp
	nosZ	PA3913	PA1051
	nirF	PA0512	speB2
	PA3284	nirJ	PA4498
	PA2662	PA3880	PA0918
	PA1746	nirQ	PA3692
	PA0515	PA3206	PA4607
	PA0512	pilW	narL
	PA3912	PA2663	cspD
	PA0510	pilV	fptA
minimal ppde	0.999999999997	0.999999960000	0.999999997000

a)



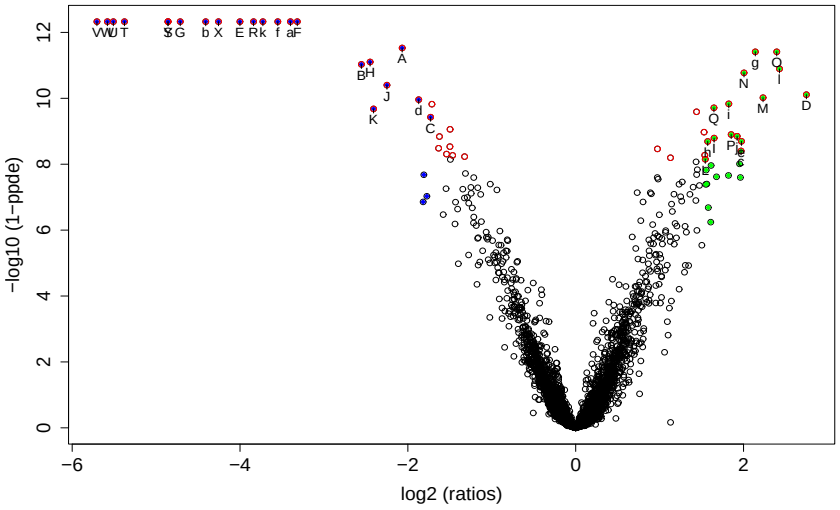
A	PA0513	K	PA2663	T	PA3871
B	nirF	L	fhp	U	narI
C	PA0521	M	PA3283	V	narJ
D	norC	N	PA3284	W	narH
E	norB	O	nosZ	X	narG
F	PA0525	P	nosD	Y	narK2
G	PA1746	Q	nosF	a	narK1
H	PA1854	R	nosY	b	moeA1
I	PA1855	S	moaA1	c	moaB1
J	PA1856				

b)



A	ig	M	norC	X	nosD
B	nirN	N	norB	Y	nosF
C	PA0510	O	PA0525	a	nosY
D	nirJ	P	PA0526	b	nosL
E	PA0512	Q	PA2128	c	narL
F	PA0513	R	PA2129	d	PA3880
G	nirL	S	PA3205	e	PA3913
H	PA0515	T	PA3283	f	PA4498
I	nirF	U	PA3284	g	PA4500
J	nirC	V	nosR	h	fimU
K	nirM	W	nosZ	i	PA4685
L	nirS				

c)



A	oprE	M	rmf	Y	narK1
B	putA	N	PA3205	a	moeA1
C	speB2	O	PA3575	b	moaB1
D	PA1746	P	PA3692	c	PA4141
E	PA1854	Q	PA3819	d	PA4205
F	PA1855	R	moaA1	e	fptA
G	PA1856	S	PA3871	f	PA4500
H	PA2128	T	narI	g	fimU
I	cspD	U	narJ	h	pilV
J	PA2663	V	narH	i	pilW
K	fhp	W	narG	j	PA4607
L	PA3040	X	narK2	k	PA4685
				l	PA4739

Figure 26: Volcano plots of the data of Benkert et al., 2008 (unpubl. res.) after preprocessing with the rma method. The x-axis is the $\log_2$ ratio (Tab. 14) and the y-axis the -$\log_{10}$-transformed 1-ppde (Tab. 15). All pairwise comparisons of the three conditions are shown: a) Wt-NO_3^- vs. Wt+NO_3^-, b) *narL*$^-$+NO_3^- vs. Wt-NO_3^- and c) *narL*$^-$ vs. Wt, both +NO_3^-. Blue and green solid circles, respectively, show the 25 genes with the lowest and highest log ratio. Red circles mark the genes with the 50 best ppde (analogous to Tab. 15). The genes belonging to the intersection of both are denoted by letters.

Table 16: Correlation of the ppde of the data of Benkert et al., 2008 (unpubl. res.) obtained for the three different preprocessing methods (analogous to Tab. 13). For each pairwise comparison of experimental conditions the Pearson's correlation coefficient was computed for all 5900 probe sets.

Compared conditions	rma vs. dChip	rma vs. vsn	vsn vs. dChip
Wt $+NO_3$ vs. Wt $-NO_3$	0.65	0.94	0.68
$narL^- + NO_3$ vs. Wt $-NO_3$	0.72	0.73	0.81
$narL^-$ vs. Wt, both $+NO_3$	0.74	0.90	0.72

a volcano plot (Jin *et al.*, 2001; Wolfinger *et al.*, 2001; Cui and Churchill, 2003; Irizarry, 2005; Chen *et al.*, 2007). Therefore, for the analyzed data sets the volcano plots were computed (Fig. 26). Additionally, the 25 genes with the highest and lowest log expression ratios, respectively, and the 50 genes with the best ppde were determined and flagged (Fig. 26). These genes are found at the upper left and upper right corners – the crests of the volcano – and are presumably differentially expressed between the two conditions analyzed, thus representing interesting candidate genes for further experiments. Several of them have been discussed above in the context of the genes with the best log ratios (Tab. 14) and best ppde (Tab. 15) such as the genes of the *narK1K2GHJI* and *norBCDFZ* operon. Overall, for the rma method the intersections between both expression measures comprised between 28 - 37 genes for the three comparisons which corresponds to 55% - 75%. Besides these genes that have both a high absolute log ratio and a high ppde, and the bulk of genes that have low values for both expression measures (located in the crater of the volcano), there lay two other groups of genes in between:

- genes with small fold-changes, but high ppde. They are located on the upper middle of the plot and are not found on the top of the list when ranking by log-ratios.
- genes with large fold-changes, but with low ppde. These genes are located on the lower left and lower right of plot. They are not found at the top of the list when ranking by ppde.

Since genes of these groups show a discrepancy between both expression measures, the log ratios and the probabilities of differential expression, they can be readily identified with the help of a volcano plot (Fig. 26). Especially those genes which have a high ppde, but scarcely fail to fulfill the 2-fold rule, can still be differentially expressed and might be biologically significant under the investigated conditions.

The next analysis, as was done above for the expression ratios (chapter 3.3.1, Tab. 13), was the determination, to which degree the ppde obtained by the three different preprocessing methods correlate. For this purpose, the correlation coefficients were computed for all three pairwise comparisons of them (Tab. 16). In general, the correlation of the methods was high ranging between 0.65 and 0.94, but weaker than that observed for the corresponding expression ratios (Tab. 13). As for the expression ratios, the correlation between the rma and vsn method was highest with one exception ($narL^- + NO_3$ vs. Wt $-NO_3$), for which that between the vsn and the dChip method was slightly higher.

Finally, in order to assess the overall similarity between both expression measures ppde and expression ratios, the Pearson's correlation coefficients were determined for all preprocessing methods (Tab. 17). In general, a medium overall correlation of 0.55 to 0.70

Table 17: Correlation of the two expression measures ppde and expression ratio as computed for the data sets of Benkert et al., 2008 (unpubl. res.). The Pearson's correlation coefficient was computed for the absolute log-ratios and the ppde of each pairwise comparison of conditions (left column) for each of the three preprocessing methods rma, dChip and vsn.

Compared Conditions	rma method	dChip method	vsn method
Wt -NO_3^- vs Wt +NO_3^-	0.55	0.69	0.57
narL$^-$ +NO_3^- vs Wt -NO_3^-	0.70	0.68	0.63
narL$^-$ vs. Wt, both +NO_3^-	0.60	0.63	0.62

was found for the three preprocessing methods. This finding supported the observations described above that genes with high absolute expression ratios often but not always have high pde, and vice versa. Therefore, for getting a comprehensive picture of the results of gene expression profiling experiment, it is beneficial to compute both expression measures, if a sufficient number of replicates – at least three or four – is available.

Altogether, the performed mid-level analysis allowed to identify genes with a high ppde and, using a volcano plot, to identify those genes which possess both a high ppde and a high absolute expression ratio. Both computed expression measures were used for the subsequent high-level analysis (chapter 3.3.3), in which the JProGO tool was used for the identification of the relevant biological functions and processes. Since for this JProGO-based analysis as much expression information as available should be used, threshold free methods were employed. Therefore, for the following high-level analysis (chapter 3.3.3) no cut-off significance level had to be determined for the ppde data, in contrast to other analyses which focus on single genes and not on whole biological processes and functions. For such single gene studies, the determination of differentially expressed genes using a predefined significance level is common (see Dudoit and Shaffer, 2003).

3.3.3 High-Level Analysis: Application of JProGO

The successful preprocessing of the data sets of Benkert et al., 2008 (unpubl. res.) with three different algorithms, the determination of the expression ratios for each and the computation of the corresponding pde were described above (chapter 3.3.1 and 3.3.2). In this chapter, these data were employed for a high-level analysis using an automatic biological interpretation with the JProGO tool. In this context, one focus was on the investigation of the influence of the preprocessing method – rma, dChip and vsn - and the type of expression data – ratios and ppde – on the outcome of the JProGO-based functional interpretation. The other focus was on the identification of the significant GO nodes for all pairwise comparisons of the three experimental conditions (see e.g. Tab. 17). The array experiment performed by Benkert et al., 2008 (unpubl.) as well as the other analyzed ones (chapter 3.3.1) constituted classical setups of the type pairwise comparison of two conditions with only a small number of two or three different experimental conditions. Therefore, they were not well suited for a data-driven analysis such as gene-based clustering. For this reason, no cluster analysis was performed and the high-level analysis was limited to the above mentioned functional interpretation with JProGO.

Both, ppde and expression ratios, were used as input data for JProGO. Again, the three analyzed pairwise comparisons of the experimental conditions comprised the wild type grown with and without NO_3^-, the *narL*$^-$ mutant grown with NO_3^- vs. the wild type grown

a)

	r(ppde)
rma vs. dChip	0.42
rma vs. vsn	0.83
dChip vs. vsn	0.43

	r(ratios)
rma vs. dChip	0.77
rma vs. vsn	0.95
dChip vs. vsn	0.76

	r(ppde vs. ratios)
rma	0.19
dChip	0.14
vsn	0.20

b)

	r(ppde)
rma vs. dChip	0.21
rma vs. vsn	0.29
dChip vs. vsn	0.26

	r(ratios)
rma vs. dChip	0.76
rma vs. vsn	0.93
dChip vs. vsn	0.76

	r(ppde vs. ratios)
rma	0.05
dChip	0.11
vsn	0.13

c)

	r(ppde)
rma vs. dChip	0.51
rma vs. vsn	0.62
dChip vs. vsn	0.45

	r(ratios)
rma vs. dChip	0.75
rma vs. vsn	0.92
dChip vs. vsn	0.71

	r(ppde vs. ratios)
rma	0.14
dChip	0.19
vsn	0.05

Figure 27: Correlation of the p-values of the GO nodes obtained by the JProGO analysis of the data of Benkert et al. (2008). The influence of the preprocessing method on the outcome of the JProGO analysis was investigated for the rma, dChip and vsn. For this purpose both, ppde (Tab. 15)and expression ratios (Tab. 14), were used for a JProGO analysis using all three pairwise comparisons of the three experimental conditions: a) Wt–NO_3^- vs. Wt+NO_3^-, b) *narL*$^-$+NO_3^- vs. Wt–NO_3^- and c) *narL*$^-$ vs. Wt, both +NO_3^-.

without NO_3^- and the *narL*$^-$ mutant vs. the wild type, both grown with NO_3^- (see e.g. Tab. 13). Nearly the same parameters of analysis were chosen as for the comparative case study in *E. coli* K-12 (chapter 3.2.2): a U-test (two-sided alternative hypothesis) using the FDR method for correcting the multiple testing effect. Only the level of significance was slightly increased up to tolerating a false discovery rate of 10%, instead of 5%, to also include GO nodes that would otherwise failed to be identified in this explorative analysis (see below). In total 12 analyses – 3 preprocessing methods × 3 condition comparisons × 2 expression data types – were performed and, in each case, p-values were obtained for almost 1000 GO nodes. A correlation analysis of these p-values was performed – a process which was independent of the level of significance – to elucidate the similarity between the JProGO results of the three preprocessing methods. For both, ppde and expression ratios, and for all three experimental pairwise combinations the correlation ($r_{Pearson}$) between the rma and the vsn method was highest ranging between 0.29 and 0.95 (Fig. 27, left and middle column). The correlation of the rma and the dChip as well as that of the vsn and the dChip method were always weaker and had almost the same value. The high correlation between the JProGO results of the rma and the vsn method fits with the higher similarity observed for the expression ratios (see Tab. 13) and the ppde of the genes (see Tab. 16). Interestingly, the correlation of the p-values of the GO nodes that were obtained with the expression ratios was in all three cases higher than those obtained for the ppde which was also observed on the level of the gene expression values (Tab. 13 and 16). For the expression ratios the smallest $r_{Pearson}$ constituted still 0.71, whereas for the ppde it was only 0.21. The results for this small number of analyses suggested that expression ratios might be the more robust type of input data for a GO-based high-level analysis than ppde regarding the correlation of different preprocessing methods. A possible explanation is that for the otherwise more robust ppde (with respect to the variance within replicate measurements) the expression values of all genes are shrinked to the interval [0:1] Since this it not the case for the expression ratios, here subgroups of

genes with high mean expression ratios can occur which would result in low and, as the case may be, significant p-values for the corresponding GO nodes. Interestingly, when using expression ratios also a higher number of significant GO nodes is found in all cases (for α=0.05 and 0.10) than with the ppde.

Altogether, the described results do not argue against the use of ppde as input data for JProGO but reflect the stronger influence of the chosen preprocessing method on the computed pde (Tab. 13 and Tab. 16), which obviously is amplified when performing a functional high-level analyses (Fig. 27). Despite of that, the agreement between the two more recent and widely used methods, rma and vsn, was not small. The JProGO results of one of them, the rma method, were presented below.

Amongst, the investigation of the overall influence of the preprocessing algorithms on the outcome of the JProGO-based functional interpretation, in the following the significant GO nodes that were identified for the analyzed data sets were presented and discussed. For this purpose and due to lack of space, the ppde obtained for the widely used rma method were chosen and the *Molecular Function* and *Biological Process* subgraphs were computed using the significant GO nodes as leaf nodes (Fig. 28, 29, 33 and 34).

For the comparison of the wild type cells grown anaerobically with and without nitrate in total 66 of the tested 962 GO nodes were identified as significant. The *Molecular Function* subgraph comprised 28 significant GO nodes and their paths up to the root node (Fig. 28). The *Biological Process* subgraph contained 32 significant nodes (Fig. 33). The more general nodes '*aerobic respiration*', '*energy derivation by oxidation of organic compounds*' and '*structural constituent of ribosome*' point to the profound adaptations in the energy metabolism of cells which were due to the withdrawal of nitrate as the terminal electron acceptor under anaerobic conditions. The underlying functions and processes could also be identified such as changes in the '*tricarboxylic acid cycle*', '*ATP synthesis coupled electron transport*', the '*NADH dehydrogenase (ubiquinone) activity*', the '*nitrate reductase activity*' and '*nitrate reductase complex*', which were represented by more specific GO nodes. The latter node is also supported by the significant node '*nitrate reductase complex*' that belongs to the *Cellular Component* subgraph (not shown) and fits with the genes with low ppde such as *narK1K2GHJI* (Tab. 15) and the results described in the literature (Krieger, 2001; Sharma *et al.*, 2006; Schreiber *et al.*, 2007). Besides NarL, which is responsible for the expression of the genes of the nitrate reductase complex, the observed alteration (see above) were obviously coordinated by several additional transcriptional regulators. This is also suggested by the presence of the significant GO nodes '*transcription factor activity*' (Fig. 28) and '*regulation of transcription, DNA-dependent*' (Fig. 33).

For the comparison of the *narL* mutant grown anaerobically with nitrate versus the wild type cells grown without nitrate no significant GO nodes were obtained for the ppde of the rma method. This was also the case for the other two preprocessing methods, vsn and dChip, and might be due to the fact that only a small number of genes is differentially expressed when comparing these two conditions.

For the comparison of the *narL* mutant versus the wild type cells, both grown anaerobically in the presence of nitrate, 21 of the analyzed 962 GO nodes were marked as significant. The *Molecular Function* subgraph comprised 9 significant GO nodes (Fig. 29) and the *Biological Process* subgraph contained 12 significant nodes (Fig. 34). The significant GO nodes constituted, basically, a subset of the nodes found for the comparison of the wild type PAO1 cells grown with versus without nitrate (see above). Likewise, as

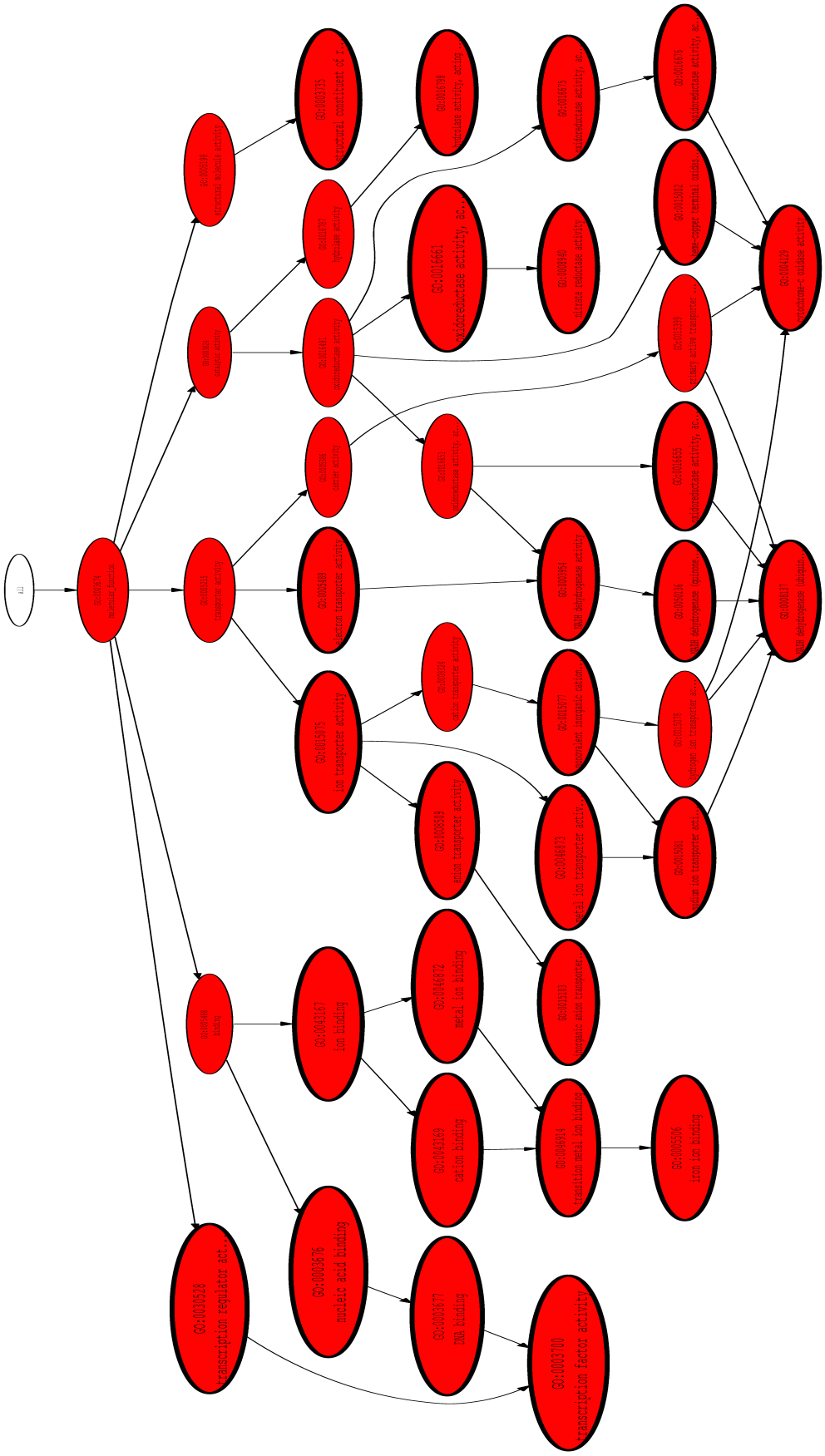


Figure 28: *Molecular Function* subgraph of the JProGO results for the expression data representing the PAO1 wild type cells grown anaerobically with versus without nitrate (Benkert et al., 2008, unpubl. res). The ppde computed for the rma-preprocessed data were used as input data for JProGO. A two-sided Mann-Whitney U-test was used with an FDR of 10%. All significant nodes and the paths up to the root node ('*all*') are present.

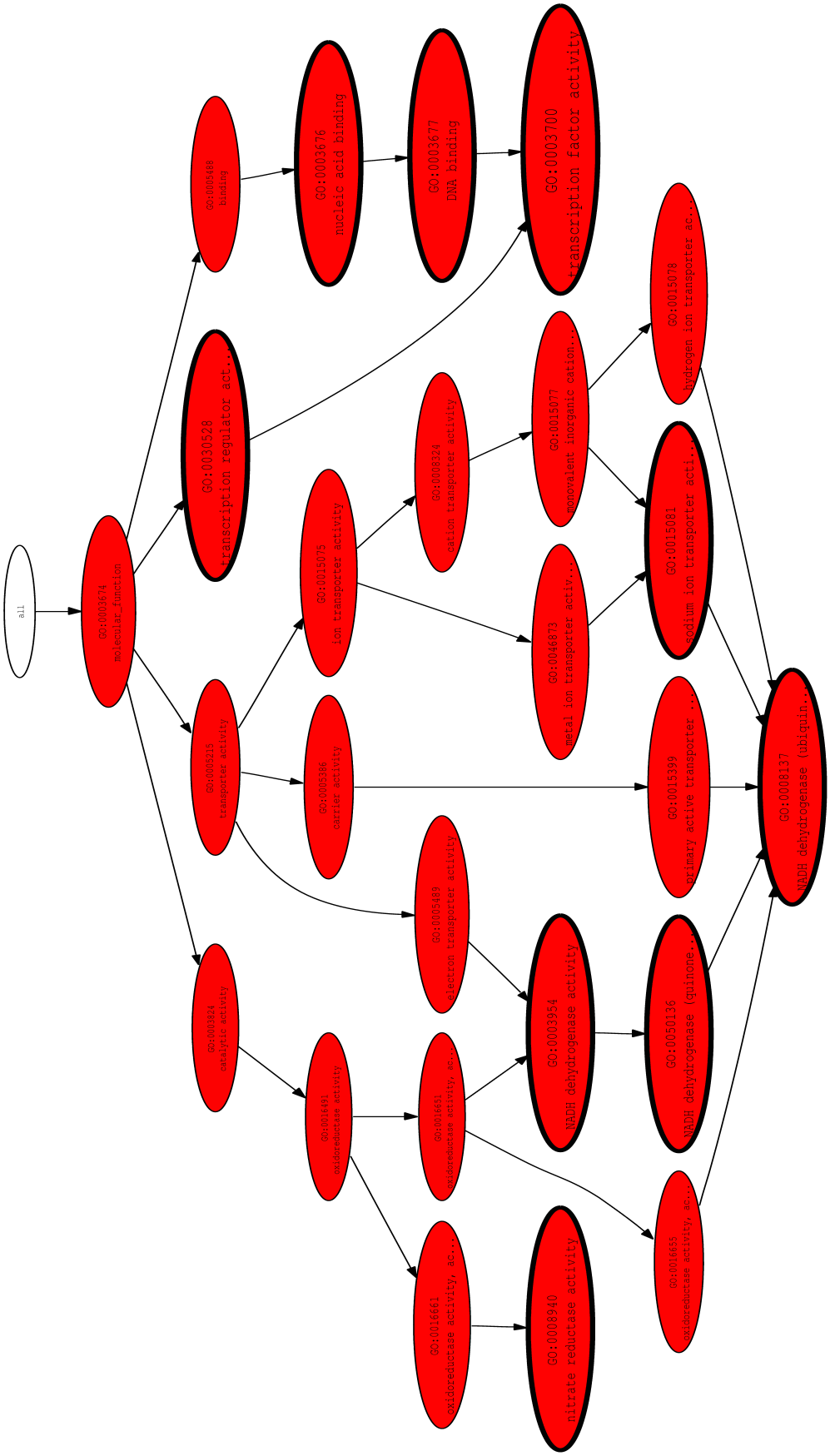


Figure 29: *Molecular Function* subgraph of the JProGO results for the expression data representing the PAO1 *narL* strain versus wild type cells grown anaerobically, both with nitrate (Benkert et al., 2008, unpubl. res). The ppde computed for the rma-preprocessed data were used as input data for JProGO. A two-sided Mann-Whitney U-test was used with an FDR of 10%. All significant nodes and the paths up to the root node ('*all*') are present.

significant GO nodes '*transcription factor activity*', '*NADH dehydrogenase (ubiquinone) activity*, '*ATP synthesis coupled electron transport*', '*regulation of transcription, DNA-dependent*' and '*nitrate reductase activity*' were found, which was consistent with the prospected functions. For example, the genes of the '*nitrate reductase activity*' would be expected to be down-regulated when their transcriptional activator NarL is not present (see above). Interestingly, the observed changes in the cellular metabolism were weaker when referring to the comparison of the wild type strain in the presence and absence of nitrate. For instance, no gross changes in the cellular energy such as '*aerobic respiration*', '*oxidative phosphorylation*' or '*energy derivation by oxidation of organic compounds*' were detected by JProGO. This argued for the presence of alternative energy recovery pathways.

Altogether, the high-level analysis of the expression data of Benkert et al., 2008 demonstrated the influence of the used preprocessing method and type of expression data on the outcome of a high-level analysis. When using expression ratios as input data type, the employed preprocessing method seemed to have a weaker effect on the results of the analysis than when using ppde. Regardless of the type of expression data the agreement between the vsn and rma method was almost always highest. Furthermore, the JProGO analysis of the analyzed expression data yielded several significant GO nodes that gave a valuable overview on the biological functions and processes that were affected under the investigated conditions.

3.4 JRegA: Expansion of the JProGO Approach Towards Regulons

3.4.1 JRegA Approach and Implemented Tool

The successful employment of JProGO in the high-level analysis of preprocessed expression data of the prokaryotic model organisms *E. coli* (chapter 3.2), *B. subtilis* (Keijser *et al.*, 2007 and chapter 3.2.3) and *P. aeruginosa* (chapter 3.3.3) was the motivation for the following approach. JProGO should be expanded beyond its initial focus, the functional classification system GO, to other types of biological groupings of genes, the regulons. A regulon represents a group of genes that is under the control of the same transcriptional regulator or stimulus (Madigan and Martinko, 2006) and, thus, the inclusion of regulons – in completion to GO – allows to consider the molecular networks underlying the regulation of gene transcription whose effects are measured by microarray-based expression profiling. Similar gene-regulatory extensions were implemented for related tools (e.g. Boorsma *et al.*, 2005), but a systematic comparison of the performance of different statistical methods, which would be desirable, was as far as known not performed up to now. Therefore, JRegA, a command line-based *Java* program prototype based on the JProGO framework, which requires a connection to *R* running in server mode (see chapter 2.11), was implemented supporting three threshold-free statistical tests (see chapter 3.1.2). Furthermore, experimentally verified regulons, including the corresponding operons, were obtained from the PRODORIC database (Münch *et al.*, 2003, 2005), which is also tightly linked to the first version of JProGO. As further extension the optional inclusion of random permutations of the expression matrix was implemented to separate random from non-random effects for each biological grouping (see also Volinia *et al.*, 2004; Barry *et al.*, 2005)). Subsequently, JRegA was applied to expression data sets investigating the knock out of transcriptional regulators (chapter 3.4.2).

3.4.2 Application of JRegA to Prokaryotic Microarray Expression Data

The JRegA tool was employed for a case study using the model organism *E. coli* for which a high number of well annotated regulons is available (see Münch *et al.*, 2005; Salgado *et al.*, 2006). Therefore, experimentally validated regulons of *E. coli* (in total 78) were extracted from the PRODORIC database (Münch *et al.*, 2003, 2005). The JRegA tool was then employed using appropriate input data. These comprised preprocessed data sets published for *E. coli* that either describe the genome-mediated knockout of a gene coding for a transcriptional regulator – such as ArcA and Fnr – or a drastic change in the cultivation conditions which affects the activity of certain global transcriptional regulators – here aerobic versus anaerobic growth. The chosen data sets are exactly those that were selected for the case study of JProGO (chapter 3.2.2 and Tab. 6) and all demands on the data sets formulated in that context, such as a sufficient number of replicates and an expectation on the outcome of the experiments (chapter 3.2.2.1), were likewise valid. Especially the last criterion of a clear expectation on the results could be much better supported for JRegA than for JProGO upon using appropriate knockout data for the following reason: When a transcriptional regulator is mutated, the genes belonging to its regulon are assumed to be mainly affected in their expression. Therefore, the corresponding JRegA groups of genes should belong to the those with the lowest p-values. Such straight forward hypotheses are, of course, not available for the biological functions and processes represented by the GO nodes in JProGO.

For JRegA the main expectations on the individual experimental conditions (see above mentioned data sets) were the following. For the comparison of either the *arcA*$^-$ strain (Salmon *et al.*, 2005), the two *fnr*$^-$ strains (Salmon *et al.*, 2005; Kang *et al.*, 2005) or the *lrp*$^-$ strain (Hung *et al.*, 2002) to the wild type strain, the corresponding regulon – ArcA, Fnr, and Lrp regulon, respectively – should be mostly affected. Since a slight Fnr-mediated regulation of the *arcA* expression had also been observed (Compan and Touati, 1994; Drapal and Sawers, 1995), in the two *fnr*$^-$ strains (Salmon *et al.*, 2005; Kang *et al.*, 2005), in addition the ArcA regulon should be influenced. For the two data sets that represent the comparison of aerobic and anaerobic growth of *E. coli* wild type cells (Salmon *et al.*, 2005; Kang *et al.*, 2005), both the ArcA regulon and the Fnr regulon were expected to obtain a low p-value since Fnr and the two-component regulatory system, ArcAB, are the two main regulatory systems that respond to decreases in the oxygen level (Gunsalus and Park, 1994; Kang *et al.*, 2005).

In the JRegA case study the same three threshold-free methods like in the JProGO case study (chapter 3.2.2.1) were used: the KS-test, the t-test and U-test. The purpose was to compare the different tests and evaluate their suitability. In addition, for each test both types of input data, pde and expression ratios, were taken in order to investigate their impact on the results of the analysis. Thus, in total six analyses were performed. The results obtained for the U-test are shown in Table 18 (pde) and Table 19 (expression ratios). The results of the two other tests can be found in the appendix (Tab. 20-23). In all cases, only regulons with at least 8 assigned genes were considered (41 regulons) and their ranks, based on their p-values, were computed.

Starting with the pde as input data type and using the expectations described above, the U-test performed well in identifying the knockout of the Fnr regulon in both *fnr*$^-$ strains, which obtained ranks 1 and 3, respectively (col. 2 and 3 of Tab. 18). For the *lrp*$^-$ strain, the Lrp regulon obtained a low rank of 4, too (col. 6 of Tab. 18). The fact that the ArcA regulon was on the first rank for both data sets that represent the growth

Table 18: *E. coli* regulons ranked by the p-values obtained for the different microarray data sets (pde) using the U-test of JRegA. Only regulons are shown to which at least 8 genes were assigned in the PRODORIC database (2^{nd} column). The order of the pairwise comparisons of experimental conditions is the same as in Tab. 6 and, if those comprise the knock out of a transcriptional regulator, the corresponding regulon row is marked in **bold**. For a better readability a horizontal bar is drawn after every fifth line. The original JRegA results (p-values) used for the computation of the ranks are shown in Table 24.

Regulon	∑Genes	*arcA*$^-$/Wt -O_2 Salmon,2005	*fnr*$^-$/Wt -O_2 Salmon,2003	*fnr*$^-$/Wt -O_2 Kang,2005	*fnr*$^-$ +O_2/-O_2 Kang,2005	*fnr*$^-$/Wt +O_2 Kang,2005	*lrp*$^-$ Wt Hung,2002	Wt +O_2/-O_2 Salmon,2003	Wt +O_2/-O_2 Kang,2005
araC	10	29	27	28	22	22	16	10	23
arcA	**69**	**27**	**31**	**5**	**1**	**10**	**34**	**1**	**1**
argR	17	12	10	35	10	9	8	8	38
caiF	10	9	5	11	41	40	18	16	25
cbl	9	14	23	23	17	13	36	23	21
cpxR	28	17	29	10	5	28	35	12	40
crp	308	1	4	29	18	18	2	3	17
cysB	17	8	41	32	12	4	30	26	24
cytR	10	21	36	36	34	41	20	24	20
dnaA	15	39	18	41	32	33	26	15	37
fadR	9	31	33	17	7	25	11	35	39
fhlA	17	6	17	31	40	5	9	30	9
fis	131	13	7	26	24	21	38	4	14
fliA	32	33	37	1	2	11	22	39	19
fnr	**62**	**4**	**1**	**3**	**38**	**1**	**5**	**14**	**3**
fruR	13	7	12	18	9	26	40	9	15
fur	26	3	9	13	13	17	24	40	4
glnG	11	36	34	25	23	34	27	31	26
glpR	8	28	28	6	30	31	12	21	6
gntR	9	15	15	21	26	30	28	13	27
lexA	87	24	11	24	33	24	31	41	29
lrp	**22**	**22**	**8**	**4**	**6**	**39**	**4**	**36**	**28**
malT	9	25	39	34	15	15	1	6	35
marA	25	18	22	12	28	16	14	28	11
metJ	35	41	19	19	21	14	15	37	34
modE	26	37	20	14	20	19	32	18	13
nagC	10	19	16	33	35	38	21	32	41
narL	75	10	38	7	11	2	19	5	2
narP	14	20	3	39	3	6	23	34	5
ompR	9	38	32	9	8	32	13	25	30
oxyR	25	2	2	16	37	37	39	38	10
phoB	30	11	25	20	25	23	25	7	33
phoP	10	23	13	27	39	35	17	11	16
purR	46	40	26	22	19	20	7	17	31
rob	19	34	35	37	36	3	33	29	18
rpoN	30	30	30	38	29	8	6	33	7
soxS	38	26	40	8	14	7	41	27	12
trpR	9	32	21	40	27	29	37	20	32
tyrR	8	35	24	30	31	27	10	19	36
yhiX	14	5	6	2	4	12	3	2	8
yiaJ	9	16	14	15	16	36	29	22	22

Table 19: *E. coli* regulons ranked by the p-values obtained for the expression ratios of different microarray data sets using the U-test of JRegA. Except for using expression ratios than ppde as input data for JRegA the table is organized exactly like Table 18. The original JRegA results (p-values) used for the computation of the ranks are shown in Table 25.

Regulon	∑Genes	$arcA^-$/Wt -O_2 Salmon,2005	fnr^-/Wt -O_2 Salmon,2003	fnr^-/Wt -O_2 Kang,2005	fnr^- +O_2/-O_2 Kang,2005	fnr^-/Wt +O_2 Kang,2005	lrp^- Wt Hung,2002	Wt +O_2/-O_2 Salmon,2003	Wt +O_2/-O_2 Kang,2005
araC	10	37	23	22	16	25	3	18	31
arcA	69	2	2	8	6	1	25	1	10
argR	17	14	8	17	37	27	22	29	14
caiF	10	20	11	21	31	28	11	11	35
cbl	9	36	24	37	32	31	30	41	41
cpxR	28	28	37	26	38	40	17	36	24
crp	308	24	38	28	13	33	1	38	36
cysB	17	41	16	40	39	39	18	19	33
cytR	10	39	41	30	28	35	26	24	38
dnaA	15	10	30	41	27	37	41	17	30
fadR	9	33	40	36	7	18	38	35	40
fhlA	17	6	9	7	25	10	7	13	4
fis	131	1	1	19	4	34	35	2	19
fliA	32	25	15	1	1	23	31	16	11
fnr	62	15	27	3	8	11	4	26	1
fruR	13	7	10	31	21	26	40	5	26
fur	26	12	18	24	5	36	15	27	7
glnG	11	38	28	34	34	3	21	39	9
glpR	8	21	21	32	15	15	12	34	32
gntR	9	16	14	20	36	7	29	21	22
lexA	87	11	4	33	24	19	37	10	29
lrp	22	13	36	10	9	17	28	40	21
malT	9	27	31	25	10	16	2	6	17
marA	25	32	22	9	33	29	14	37	25
metJ	35	40	32	15	41	20	34	25	18
modE	26	30	39	5	12	24	27	30	13
nagC	10	18	26	29	23	21	23	14	28
narL	75	31	19	2	14	2	10	7	2
narP	14	17	12	18	2	4	20	28	3
ompR	9	34	20	16	18	6	39	32	37
oxyR	25	29	33	11	29	9	24	31	5
phoB	30	9	13	35	11	32	6	4	20
phoP	10	8	6	14	40	22	19	22	12
purR	46	22	17	38	26	30	33	12	34
rob	19	19	29	13	17	13	36	23	15
rpoN	30	35	35	12	22	12	9	15	16
soxS	38	4	3	6	35	8	5	20	6
trpR	9	5	5	39	20	41	16	8	27
tyrR	8	26	34	23	19	38	32	33	39
yhiX	14	3	25	4	3	5	8	3	8
yiaJ	9	23	7	27	30	14	13	9	23

of wild type cells under aerobic versus and anaerobic conditions (col. 7 and 8 of Tab. 18) was also in good agreement with the expectation. In addition, for one of the two data sets the other O_2-responsive regulon, the Fnr regulon, had a low rank of 3 (col. 8 of Tab. 18). The high rank (place 38 out of 41) of the Fnr regulon that was computed for the comparison of the *fnr*$^-$ strain grown under aerobic and anaerobic conditions fits well with the expectation, too, since no transcriptional differences in this regulon should be observed through the knockout of the corresponding regulator. The low rank of 1 for this regulon obtained, on the other hand, in the comparison of the *fnr*$^-$ strain and the wild type strain under aerobic conditions (col. 5 of Tab. 18), might be due to the fact that Fnr still shows a weak activity even under higher O_2 levels. Altogether, the results of the U-test obtained with the pde shows a good agreement with the expected regulons with solely one exception, the *arcA*$^-$ strain: In this case the corresponding ArcA regulon only obtained a comparatively poor rank of 27 (col. 1 of Tab. 18), and here the results computed for the expression ratios were clearly better yielding a rank of 2 (Tab. 19).

The two other statistical tests, the KS-test and the t-test, gave similar results (Tab. 20 and 21): With respect to the above formulated expectations, the KS-test performed even better than the U-test in almost all cases and the t-test was worst. The KS-test assigned a better rank, for example, to the ArcA regulon (rank 17 instead of three 27) and in one of the two cases for the Fnr regulon (rank 2 instead of 3) when using the respective knockout data set (col. 1,3 and 6 of Tab. 20). Likewise, the Lrp regulon obtained a rank 2 instead of 4.

After using the pde as input data for JRegA, in the following the expression ratios are used to evaluate the performance of the three statistical tests in comparison to the pde. When taking the theoretical considerations in to account and the findings from the JProGO case study (chapter 3.2.2), the expression ratios should be more susceptible to fluctuations in the replicate measurements and other experimental noise of the measured array data. The application of JRegA to expression ratios seemed to yield, indeed, poorer results as compared to those of the pde (Tab. 19, 22 and 23): When looking again at the U-test, most of the regulons that were expected to have low p-values get considerably higher, thus poorer, ranks than with the pde. For example, with one of the fnr$^-$ strains (Kang *et al.*, 2005), the Fnr regulon gets only rank 27 instead of 3 (col. 2 of Tab. 19), with the lrp$^-$ strain the Lrp regulon reaches only rank 28 instead of 4 (col. 6 of Tab. 19) and the ArcA regulon gets with the anaerobically grown wild type strain of Kang *et al.* (2005) only rank 10 instead of 1 was assigned (col. 8 of Tab. 19). On the other hand, the expression ratios perform in one case better than the pde: the ArcA regulon of the arcA$^-$ strain obtained – as expected – a quite low rank of 2 instead of 27. Similar observations were made when comparing the JRegA outputs for the pde and expression ratios for the two other statistical tests (compare Tab. 20 to 22 and Tab. 21 to 23), whereas from all three methods the t-test showed the greatest decline and the KS-test the weakest. In addition, as for the pde, the KS-test performs best with the expression ratios when referring to the agreement with the above formulated expectations.

Altogether, similar to JProGO, the pde were found the more appropriate type of expression data for a JRegA analysis. This might, amongst the above mentioned reasons, at least partially be due to the fact that transcriptional changes of the genes in a regulon normally occur in both directions comprising activation and repression. Thus, the respective expression ratios are both, greater and less than one, and so the mean expression ratio can often be near one. Through this, especially in the case of tests with mean-based test statistics, such as the t-test, the mean expression ratio of the genes belonging to the

regulon node and that of the genes that do not belong to it, might be similar, which would produce a poor p-value. This is not be the case for the pde, where the mean pde of differentially expressed genes belonging to the regulon node – regardless whether up- or down-regulated – remains clearly less than one. This effect could also explain the poorer performance of the t-test in comparison to the U-test and KS-test when using expression ratios. A possible solution, which in addition would allow a more refined analysis, would comprise the creation of sub-regulons that either contain only the up- or the down-regulated genes. The inferiority of the t-test compared to the other two tests that was observed for the pde as input data, might be explained by the fact that the pde are not normally distributed neither of the genes belonging to a particular regulon nor those that do not. Since this is one of the prerequisites of the t-test, its power might be lower leading to false positive hits (Zöfel, 2002).

Finally, on the basis of the acquired results the following conclusions were drawn:

- the JRegA snapshot analysis shows the successful expansion of the JProGO approach towards identifying transcriptionally altered regulons in the high-level analysis of prokaryotic expression data
- for a JRegA analysis the use of pde rather than expression ratios is recommended
- as statistical test the KS-test should be preferred, whereas the U-test yields almost as good results and the t-test performs worst

In general, the results obtained in the case study described for *E. coli* were promising despite the unavoidable lack of information on all experimentally verified genes that belong to a regulon. Due to its good performance, the presented program prototype JRegA should be made available to the scientific community as a full web-based service like JProGO. First steps towards the implementation of a web-based version (`http://www.jprogo.de/JRegA`) were already taken. For the creation of a stable productive program version, the following further features should be included and issues should be addressed: An appropriate correction for the multiple testing effect, preferentially a permutation-based approach similar to that of Volinia *et al.* (2004), should be implemented to compute, besides the ranks determined here, a meaningful p-value for each regulon. Furthermore, sub-regulons could be created representing genes of the regulon that are either positively or negatively regulated by the transcriptional regulator (see above). One strength of the JRegA approach is the tight coupling to a rich source of experimentally verified regulons of many prokaryotic species, the PRODORIC database (Münch *et al.*, 2003, 2005). Since due to the experimental expense only a small fraction of the members of a regulon are known, a contextual expansion towards including, in addition, predicted regulons such as obtained by the *VirtualFootprint* tool, a module of the new PRODORIC framework (Münch *et al.*, 2005) would be desirable. The regulon genes predicted this way could then be combined with an accurate operon prediction tool (see e.g. Price *et al.*, 2005; Westover *et al.*, 2005; Jacob *et al.*, 2005) to identify other member genes. It will be interesting to see whether and to which extent the employment of the predicted regulon data could further improve the performance of JRegA.

4 Conclusions and Outlook

4.1 Conclusions

JProGO Software

With JProGO a freely accessible web-based program suite was established, which enables the intuitive functional interpretation of high-throughput gene expression data using the Gene Ontology (GO) as classification system. It allows the straightforward analysis of expression data from more than 20 different prokaryotic species, including the model organisms *E. coli* (K-12) and *B. subtilis* (strain 168) as well as many medically relevant pathogens. No preparatory steps such as generating the gene-to-GO node assignment are required, because these were precomputed. Since prokaryotic expression data are rarely supported by related tools (see Blom *et al.*, 2007), JProGO offers the use of the most common statistical algorithms and methods of correction for the multiple testing effect. The acceptance by the target user group, the microbiology community, is underlined by the usage of JProGO e.g. in a time series analysis of *B. subtilis* (see Keijser *et al.*, 2007). Furthermore, the web access statistics of the JProGO server revealed analyses performed from more than 100 distinct IP addresses between June 2007 and February 2008. In comparison to related free tools, which normally provide only a single statistical test for the identification of significant GO nodes, the intention of JProGO is to provide various threshold-based or threshold-free methods. Thus, JProGO creates the prerequisites for a comparative analysis of different statistical algorithms. The latter is facilitated by the possibility of using several different statistical tests for the analysis of the same expression data set under otherwise identical conditions. This comprises the recognition of alternative gene names and the method of correction for the multiple testing effect. The outlined features were exploited in several comparative case studies. As far as known, this was one of the most comprehensive analyses of this type.

The first finding was that all statistical tests, the threshold-based Fisher's exact test as well as the threshold-free Student's t-, Kolmogorov-Smirnov- and unpaired Wilcoxon test, are able to identify relevant GO nodes that fit with the biological expectation. However, as expected, the p-values assigned to the GO nodes, the resulting rank order and the number of identified nodes varied between the methods. Despite the observed overlap between Fisher's exact test and the U-test, threshold-dependent tests in general have the severe disadvantage of the arbitrariness of the cut-off value. This cut-off and its strong impact on the outcome of the analysis was also confirmed for Fisher's exact test. Therefore, only the threshold-free tests were assessed in detail with respect to their properties and performance. In this context, as far as one can make generalizations from the limited number of expression data sets from *E. coli*, the following tendency arises: The KS-test is comparatively selective. It reliably identifies a subset of relevant GO nodes, but several others are missed that were found by the other two threshold-free tests. The t-test represents to some degree the other extreme, since it marked in each case most nodes as significant using the tested data sets. However, not all of the found nodes are biologically relevant and, on the other hand, not all important nodes, that were found by the other two cut off-free tests, are identified. Since the basis for the t-test, the existence of two normal-like distributions, is not always fulfilled, it can bear a lower statistical power leading to the observed erroneous identification of nodes that obviously constitute false-positive hits. The GO node-specific view of JProGO, which enables the visualization of the two expression value distributions of a) the genes assigned

a particular GO node and b) that of the remaining genes, could help to decide whether the data are normally distributed and the application of the t-test is justified. The last evaluated threshold-free test, the U-test, lies in between the other two. It bears the potential to identify more biological meaningful nodes than the KS-test (see case study with expression data from *E. coli*), while keeping the portion of false-positives smaller than the t-test. This finding emphasizes the aptitude of this non-parametric rank-based test for the analysis of expression data, whereas artificial data sets with extensively shared ranks might be generated for which the other two statistical tests would perform better. The good performance of the U-test in the JRegA case study (see below), in which a clear expectation on the outcome of the analysis exists, indicates its general suitability for the functional interpretation of prokaryotic expression data using different biological groupings such as GO or regulons. In this context, the KS-test constitutes an acceptable alternative and to some extent this holds also true for the t-test.

JRegA – Expansion of the JProGO approach towards regulons
The JRegA project demonstrates the successful extension of the functional interpretation of expression data, as offered by JProGO, from GO terms towards other biological grouping of genes. Here, regulons were used which constitute the building block of gene-regulatory networks. Appropriate data sets from knockout strains of transcriptional regulators offered, in comparison to the GO terms, the direct biological evaluation of the obtained results against the background of the affected regulons. In this context, usage of the threshold-free KS-test can be recommended, whereas the U-test yields similarly good results. Altogether, the JRegA expansion constitutes a useful supplement to the JProGO framework.

4.2 Outlook

In the future, JProGO should be expanded towards containing further prokaryotic strains and, ideally, to include all sequenced bacterial and archaeal genomes, which could for example be obtained from the PRODORIC database. Moreover, the program could also be extended to support the analysis of more than two experimental conditions (see the pioneering work of Zeeberg *et al.*, 2005), which would also imply to offer additional statistical methods that can cope with these requirements, e.g. the analysis of variance (ANOVA). In addition, the implementation of permutation-based procedures for the identification of the significant GO nodes is planned. Permutation-based approaches (see e.g. Volinia *et al.*, 2004) are well suited for correcting multiple dependent tests and, therefore, constitute an appealing alternative to the Bonferroni and FDR method. Because permutations require longer running times, JProGO should provide the feature of an asynchronous web-based service, which allows to send the obtained results to the user, when the analysis has finished. Furthermore, an exploratory data mining strategy, which combines results from the different threshold-free methods, could represent an interesting alternative for JProGO. This would especially hold true, when the user places value on reducing the risk of obtaining false-positives and would like to focus on a few highly confident GO nodes: At the first step, only the intersection of significant nodes of all three threshold-based methods could be considered. Additional nodes, that are solely found by one or two of the methods, could be taken into account afterwards. Inspired by the investigation of the impact of the type of expression data, use of test statistics such as probabilities of differential expression, if sufficient replicates are available, is recommended. On the other hand, a future expansion of JProGO could comprise the establishment of a combined analysis that integrates both expression ratios and test statistics. For this purpose, a scoring function which incorporates both types of expression data, a Bayesian approach or some kind of multidimensional statistical test would be appropriate. Moreover, the influence of the employed preprocessing methods on the outcome of the high-level analysis, should not be underestimated. Despite the observed partial high correlation between the computed gene expression levels (e.g. ppde), the correlation of the p-values obtained for the GO nodes seems generally to be weaker.

Interesting innovations for the JRegA tool could be – besides the experimentally verified target genes of transcription factors, which only represent parts of a regulon – the inclusion of predicted regulon genes and the introduction of sub-regulons containing genes regulated either positively or negatively by the same transcription factor. This information could be obtained from the PRODORIC database (Münch *et al.*, 2003, 2005) and by applying the regulon prediction tool Virtual Footprint (Münch *et al.*, 2005). PRODORIC is already the main information source of experimentally verified regulons for JRegA. The next steps of development could include the combined high-level analysis using both, GO terms and regulons, as well as further biological groupings of genes or gene products such as physically interacting proteins or members of the same signal transduction pathways. By doing so, a characteristic fingerprint of the biological functions and affected regulatory sub-networks could be obtained in a comprehensive case study for a variety of transcription factor knockout data sets. This could be accompanied by a simultaneous visualization of the biological functions and networks. Finally, another application based on JRegA is conceivable: the prediction or expansion of existing regulons using respective knockout data sets (compare to the operon prediction approach with expression data by Steinhauser *et al.*, 2004). Here, the statistical test could serve as kind of scoring function.

5 Abbreviations and Glossary

Abbreviation	Whole Phrase
API	application programming interface
bp	base pairs
BP	biological process
CC	cellular component
CPU	central processing unit
DAG	directed acyclic graph
DBMS	database management system
DDBJ	DNA Data Bank of Japan
DNA	deoxyribonucleic acid
EBI	European Bioinformatics Institute
EMBL	European Molecular Biology Laboratory
GO	Gene Ontology
GOA	Gene Ontology annotation
JProGO	*Java*-based tool for the functional analysis of prokaryotic microarray data using the Gene Ontology
JRegA	*Java*-based tool for the regulon analysis of expression data
JSP	*Java server page/pages*
KS	Kolmogorov-Smirnov
KS-test	Kolmogorov-Smirnov test
LB medium	lysogeny broth medium
mRNA	messenger ribonucleic acid
MF	molecular function
MM	mismatch
NCBI	National Center for Biotechnology Information
OLN	ordered locus name
ORF	open reading frame
PCR	polymerase chain reaction
PM	perfect match
pde	probability/-ies of differential expression
ppde	posterior probability/-ies of differential expression
PRODORIC	prokaryotic database of gene regulation
RAM	random access memory
RNA	ribonucleic acid
rRNA	ribosomal ribonucleic acid
RT-PCR	reverse transcription polymerase chain reaction
SQL	Structured Query Language
TFBS	transcription factor binding site/-es
TRN	transcriptional regulatory network
tRNA	transfer ribonucleic acid
t-test	Student's t-test
TU	transcription unit
U	Mann-Whitney U
U-test	Mann-Whitney U-test
URL	universal resource locator
Wt	wild type strain
XML	extensible markup language

References

Affymetrix Inc. (2001). Statistical algorithms reference guide. Technical report, Affymetrix, Santa Clara, CA, USA.

Affymetrix Inc. (2002). Statistical algorithms description document. Technical report, Affymetrix, Santa Clara, CA, USA.

Aguilar-Mahecha, A., Hassan, S., Ferrario, C. and Basik, M. (2006). Microarrays as validation strategies in clinical samples: tissue and protein microarrays. *OMICS*, **10**, 311 – 326.

Al-Shahrour, F., Diaz-Uriarte, R. and Dopazo, J. (2004). FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes. *Bioinformatics*, **20**, 578 – 580.

Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K. and Watson, J. D. (1994). *Molecular Biology of the Cell*. Garland Publishing, Inc., 3. edition.

Allen, F. H. and Taylor, R. (2004). Research applications of the cambridge structural database (csd). *Chem Soc Rev*, **33**, 463 – 475.

Allison, D. B., Cui, X., Page, G. P. and Sabripour, M. (2006). Microarray data analysis: from disarray to consolidation and consensus. *Nat Rev Genet*, **7**, 55 – 65.

Alon, U. (2003). Biological networks: the tinkerer as an engineer. *Science*, **301**, 1866 – 1867.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. and Lipman, D. J. (1990). Basic local alignment search tool. *J Mol Biol*, **215**, 403 – 410.

Arnold, C. N., McElhanon, J., Lee, A., Leonhart, R. and Siegele, D. A. (2001). Global analysis of Escherichia coli gene expression during the acetate-induced acid tolerance response. *J Bacteriol*, **183**, 2178 – 2186.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M. and Sherlock, G. (2000). Gene ontology: tool for the unification of biology. *Nat Genet*, **25**, 25 – 29.

Baldi, P. and Long, A. D. (2001). A Bayesian framework for the analysis of microarray expression data: regularized t -test and statistical inferences of gene changes. *Bioinformatics*, **17**, 509 – 519.

Ball, C. A., Awad, I. A. B., Demeter, J., Gollub, J., Hebert, J. M., Hernandez-Boussard, T., Jin, H., Matese, J. C., Nitzberg, M., Wymore, F., Zachariah, Z. K., Brown, P. O. and Sherlock, G. (2005). The stanford microarray database accommodates additional microarray platforms and data formats. *Nucleic Acids Res*, **33**, D580 – D582.

Barabasi, A.-L. and Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. *Nat Rev Genet*, **5**, 101 – 113.

Barrett, T., Suzek, T. O., Troup, D. B., Wilhite, S. E., Ngau, W.-C., Ledoux, P., Rudnev, D., Lash, A. E., Fujibuchi, W. and Edgar, R. (2005). NCBI GEO: mining millions of expression profiles–database and tools. *Nucleic Acids Res*, **33**, D562 – D566.

Barrett, T., Troup, D. B., Wilhite, S. E., Ledoux, P., Rudnev, D., Evangelista, C., Kim, I. F., Soboleva, A., Tomashevsky, M. and Edgar, R. (2007). NCBI GEO: mining tens of millions of expression profiles–database and tools update. *Nucleic Acids Res*, **35**, D760 – D765.

Barry, W. T., Nobel, A. B. and Wright, F. A. (2005). Significance analysis of functional categories in gene expression studies: a structured permutation approach. *Bioinformatics*, **21**, 1943 – 1949.

Barthelmes, J., Ebeling, C., Chang, A., Schomburg, I. and Schomburg, D. (2007). BRENDA, AMENDA and FRENDA: the enzyme information system in 2007. *Nucleic Acids Res*, **35**, D511 – D514.

Battista, G. D., Eades, P., Tamassia, R. and Tollis, I. G. (1994). Algorithms for drawing graphs: an annotated bibliography. *Computational Geometry*, **4**, 235 – 282.

Ben-Shaul, Y., Bergman, H. and Soreq, H. (2005). Identifying subtle interrelated changes in functional gene categories using continuous measures of gene expression. *Bioinformatics*, **21**, 1129 – 1137.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *J Roy Statistical Society*, **57**, 289 – 300.

Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, **29**, 1165 – 1188.

Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J. and Wheeler, D. L. (2007). GenBank. *Nucleic Acids Res*, **35**, D21 – D25.

Berman, H., Henrick, K., Nakamura, H. and Markley, J. L. (2007). The worldwide protein data bank (wwpdb): ensuring a single, uniform archive of pdb data. *Nucleic Acids Res*, **35**, D301 – D303.

Berriz, G. F., King, O. D., Bryant, B., Sander, C. and Roth, F. P. (2003). Characterizing gene sets with funcassociate. *Bioinformatics*, **19**, 2502 – 2504.

Bickel, D. R. (2004). Degrees of differential gene expression: detecting biologically significant expression differences and estimating their magnitudes. *Bioinformatics*, **20**, 682 – 688.

Bland, J. M. and Altman, D. G. (1995). Multiple significance tests: the Bonferroni method. *BMJ*, **310**, 170.

Blom, E.-J., Bosman, D. W. J., van Hijum, S. A. F. T., Breitling, R., Tijsma, L., Silvis, R., Roerdink, J. B. T. M. and Kuipers, O. P. (2007). Fiva: Functional information viewer and analyzer extracting biological knowledge from transcriptome data of prokaryotes. *Bioinformatics*, **23**, 1161 – 1163.

Bolstad, B., Collin, F., Brettschneider, J., Simpson, K., Cope, L., Irizarry, R. and Speed, T. (2005a). *Quality assessment of affymetrix GeneChip data.*, chapter 3. Springer, Inc., 33 – 47.

Bolstad, B., Irizarry, R., Gautier, L. and Wu, Z. (2005b). *Preprocessing high-density oligonucleotide arrays.*, chapter 2. Springer, Inc., 13 – 32.

Bonferroni, C. (1936). Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, **8**, 1 – 62.

Boorsma, A., Foat, B. C., Vis, D., Klis, F. and Bussemaker, H. J. (2005). T-profiler: scoring the activity of predefined groups of genes using gene expression data. *Nucleic Acids Res*, **33**, W592 – W595.

Boyle, E. I., Weng, S., Gollub, J., Jin, H., Botstein, D., Cherry, J. M. and Sherlock, G. (2004). GO::TermFinder–open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. *Bioinformatics*, **20**, 3710 – 3715.

Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C. A., Causton, H. C., Gaasterland, T., Glenisson, P., Holstege, F. C., Kim, I. F., Markowitz, V., Matese, J. C., Parkinson, H., Robinson, A., Sarkans, U., Schulze-Kremer, S., Stewart, J., Taylor, R., Vilo, J. and Vingron, M. (2001). Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nat Genet*, **29**, 365 – 371.

Brazma, A., Parkinson, H. and ArrayExpress team, E.-E. (2006). Arrayexpress service for reviewers/editors of dna microarray papers. *Nat Biotechnol*, **24**, 1321 – 1322.

Breitling, R., Amtmann, A. and Herzyk, P. (2004). Iterative group analysis (iga): a simple tool to enhance sensitivity and facilitate interpretation of microarray experiments. *BMC Bioinformatics*, **5**, 34.

Breslin, T., Edén, P. and Krogh, M. (2004). Comparing functional annotation analyses with catmap. *BMC Bioinformatics*, **5**, 193.

Brinkman, A. B., Ettema, T. J. G., de Vos, W. M. and van der Oost, J. (2003). The lrp family of transcriptional regulators. *Mol Microbiol*, **48**, 287 – 294.

Camon, E., Magrane, M., Barrell, D., Binns, D., Fleischmann, W., Kersey, P., Mulder, N., Oinn, T., Maslen, J., Cox, A. and Apweiler, R. (2003). The Gene Ontology Annotation (GOA) project: implementation of GO in SWISS-PROT, TrEMBL, and InterPro. *Genome Res*, **13**, 662 – 672.

Camon, E., Magrane, M., Barrell, D., Lee, V., Dimmer, E., Maslen, J., Binns, D., Harte, N., Lopez, R. and Apweiler, R. (2004). The gene ontology annotation (GOA) database: sharing knowledge in uniprot with gene ontology. *Nucleic Acids Res*, **32**, D262 – D266.

Carmona-Saez, P., Pascual-Marqui, R. D., Tirado, F., Carazo, J. M. and Pascual-Montano, A. (2006). Biclustering of gene expression data by Non-smooth Non-negative Matrix Factorization. *BMC Bioinformatics*, **7**, 78.

Causton, H., Quackenbush, J. and Brazma, A. (2003). *Microarray Gene Expression Data Analysis: A Beginner's Guide*. Blackwell Publishing, Incorporated.

Chaudhuri, J. D. (2005). Genes arrayed out for you: the amazing world of microarrays. *Med Sci Monit*, **11**, 52 – 62.

Chee, M., Yang, R., Hubbell, E., Berno, A., Huang, X. C., Stern, D., Winkler, J., Lockhart, D. J., Morris, M. S. and Fodor, S. P. (1996). Accessing genetic information with high-density dna arrays. *Science*, **274**, 610 – 614.

Chen, J. J., Tsai, C.-A., Tzeng, S. and Chen, C.-H. (2007). Gene selection with multiple ordering criteria. *BMC Bioinformatics*, **8**, 74.

Cheng, Y. and Church, G. M. (2000). Biclustering of expression data. *Proc Int Conf Intell Syst Mol Biol*, **8**, 93 – 103.

Choi, C., Münch, R., Leupold, S., Klein, J., Siegel, I., Thielen, B., Benkert, B., Kucklick, M., Schobert, M., Barthelmes, J., Ebeling, C., Haddad, I., Scheer, M., Grote, A., Hiller, K., Bunk, B., Schreiber, K., Retter, I., Schomburg, D. and Jahn, D. (2007). Systomonas–an integrated database for systems biology analysis of pseudomonas. *Nucleic Acids Res*, **35**, D533 – D537.

Churchill, G. A. (2002). Fundamentals of experimental design for cDNA microarrays. *Nat Genet*, **32**, 490 – 495.

Compan, I. and Touati, D. (1994). Anaerobic activation of arcA transcription in *Escherichia coli*: roles of Fnr and ArcA. *Mol Microbiol*, **11**, 955 – 964.

Conway, T. and Schoolnik, G. K. (2003). Microarray expression profiling: capturing a genome-wide portrait of the transcriptome. *Mol Microbiol*, **47**, 879 – 889.

Cope, L. M., Irizarry, R. A., Jaffee, H. A., Wu, Z. and Speed, T. P. (2004). A benchmark for Affymetrix GeneChip expression measures. *Bioinformatics*, **20**, 323 – 331.

Cui, X. and Churchill, G. A. (2003). Statistical tests for differential expression in cdna microarray experiments. *Genome Biol*, **4**, 210.

Deutscher, M. P. (2006). Degradation of RNA in bacteria: comparison of mRNA and stable RNA. *Nucleic Acids Res*, **34**, 659 – 666.

Dharmadi, Y. and Gonzalez, R. (2004). DNA microarrays: experimental issues, data analysis, and application to bacterial systems. *Biotechnol Prog*, **20**, 1309 – 1324.

Doniger, S. W., Salomonis, N., Dahlquist, K. D., Vranizan, K., Lawlor, S. C. and Conklin, B. R. (2003). Mappfinder: using gene ontology and genmapp to create a global gene-expression profile from microarray data. *Genome Biol*, **4**, R7.

Dopazo, J. (2006). Functional interpretation of microarray experiments. *OMICS*, **10**, 398 – 410.

Drapal, N. and Sawers, G. (1995). Promoter 7 of the *Escherichia coli pfl* operon is a major determinant in the anaerobic regulation of expression by ArcA. *J Bacteriol*, **177**, 5338 – 5341.

Dudoit, S. and Shaffer, C. B. J. (2003). Multiple hypothesis testing in microarray experiments. *Statistical Science*, **18**, 71 – 103.

Duggan, D. J., Bittner, M., Chen, Y., Meltzer, P. and Trent, J. M. (1999). Expression profiling using cdna microarrays. *Nat Genet*, **21**, 10 – 14.

Eisen, M. B., Spellman, P. T., Brown, P. O. and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A*, **95**, 14863 – 14868.

Eymann, C., Homuth, G., Scharf, C. and Hecker, M. (2002). Bacillus subtilis functional genomics: global characterization of the stringent response by proteome and transcriptome analysis. *J Bacteriol*, **184**, 2500 – 2520.

Gansner, E., Koutsofios, E. and North, S. (2002). Drawing graphs with dot. Technical report, AT&T Bell Laboratories, Murray Hill, NJ, USA.

Gansner, E., Koutsofios, E. and North, S. (2006). Drawing graphs with dot. Technical report, AT&T Bell Laboratories, Murray Hill, NJ, USA.

Gautier, L., Cope, L., Bolstad, B. M. and Irizarry, R. A. (2004). Affy–analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics*, **20**, 307 – 315.

Gene Ontology Consortium (2001). Creating the gene ontology resource: design and implementation. *Genome Res*, **11**, 1425 – 1433.

Gene Ontology Consortium (2006). The Gene Ontology (GO) project in 2006. *Nucleic Acids Res*, **34**, D322 – D326.

Gentleman, R., Carey, V., Huber, W., Irizarry, R. and Dudoit, S., editors (2005). *Bioinformatics and Computational Biology Solutions Using R and Bioconductor (Statistics for Biology and Health)*. Springer, Inc., Secaucus, NJ, USA.

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., Hornik, K., Hothorn, T., Huber, W., Iacus, S., Irizarry, R., Leisch, F., Li, C., Maechler, M., Rossini, A. J., Sawitzki, G., Smith, C., Smyth, G., Tierney, L., Yang, J. Y. H. and Zhang, J. (2004). Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol*, **5**, R80.

Gruber, T. (2008). Ontology. In L. Liu and M. T. Özsu, editors, *Encyclopedia of Database Systems*. Springer, Inc.

Gruber, T. R. (1993). Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Publishers, Deventer, The Netherlands.

Gunsalus, R. P. and Park, S. J. (1994). Aerobic-anaerobic gene regulation in *Escherichia coli*: control by the arcab and fnr regulons. *Res Microbiol*, **145**, 437 – 450.

Hansen, A. (2001). *Bioinformatics – A guideline for natural scientists*. Birkhäuser (Basel, Boston, Berlin).

Hardiman, G. (2004). Microarray platforms–comparisons and contrasts. *Pharmacogenomics*, **5**, 487 – 502.

Harris, M. A., Clark, J., Ireland, A., Lomax, J., Ashburner, M., Foulger, R., Eilbeck, K., Lewis, S., Marshall, B., Mungall, C., Richter, J., Rubin, G. M., Blake, J. A., Bult, C., Dolan, M., Drabkin, H., Eppig, J. T., Hill, D. P., Ni, L., Ringwald, M., Balakrishnan, R., Cherry, J. M., Christie, K. R., Costanzo, M. C., Dwight, S. S., Engel, S., Fisk, D. G., Hirschman, J. E., Hong, E. L., Nash, R. S., Sethuraman, A., Theesfeld, C. L., Botstein, D., Dolinski, K., Feierbach, B., Berardini, T., Mundodi, S., Rhee, S. Y., Apweiler, R., Barrell, D., Camon, E., Dimmer, E., Lee, V., Chisholm, R., Gaudet, P., Kibbe, W., Kishore, R., Schwarz, E. M., Sternberg, P., Gwinn, M., Hannick, L., Wortman, J., Berriman, M., Wood, V., de la Cruz, N., Tonellato, P., Jaiswal, P., Seigfried, T., White, R. and Consortium, G. O. (2004). The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res*, **32**, D258 – D261.

Hatfield, G. W., Hung, S.-P. and Baldi, P. (2003). Differential analysis of DNA microarray gene expression data. *Mol Microbiol*, **47**, 871 – 877.

Hocquette, J. F. (2005). Where are we in genomics? *J Physiol Pharmacol*, **56**, 37 – 70.

Holtmann, G., Bakker, E. P., Uozumi, N. and Bremer, E. (2003). KtrAB and KtrCD: two K+ uptake systems in *Bacillus subtilis* and their role in adaptation to hypertonicity. *J Bacteriol*, **185**, 1289 – 1298.

Huber, W., Li, X. and Gentleman, R. (2005). *Visualizing data.*, chapter 10. Springer, Inc.

Huber, W., von Heydebreck, A., Sueltmann, H., Poustka, A. and Vingron, M. (2003). Parameter estimation for the calibration and variance stabilization of microarray data. *Stat Appl Genet Mol Biol*, **2**, Article 3.

Huber, W., von Heydebreck, A., Sultmann, H., Poustka, A. and Vingron, M. (2002). Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, **18**, S96 – S104.

Hung, S., Baldi, P. and Hatfield, G. W. (2002). Global gene expression profiling in Escherichia coli K12. The effects of leucine-responsive regulatory protein. *J Biol Chem*, **277**, 40309 – 40323.

Ideker, T., Ozier, O., Schwikowski, B. and Siegel, A. F. (2002). Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, **18 Suppl 1**, S233 – S240.

Ideker, T., Thorsson, V., Siegel, A. F. and Hood, L. E. (2000). Testing for differentially-expressed genes by maximum-likelihood analysis of microarray data. *J Comput Biol*, **7**, 805 – 817.

Irizarry, R. (2005). *From CEL files to annotated list of interesting genes.*, chapter 25. Springer, Inc.

Irizarry, R. A., Bolstad, B. M., Collin, F., Cope, L. M., Hobbs, B. and Speed, T. P. (2003a). Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Res*, **31**, e15.

Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U. and Speed, T. P. (2003b). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, **4**, 249 – 264.

Irizarry, R. A., Wu, Z. and Jaffee, H. A. (2006). Comparison of Affymetrix GeneChip expression measures. *Bioinformatics*, **22**, 789 – 794.

Jacob, E., Sasikumar, R. and Nair, K. N. R. (2005). A fuzzy guided genetic algorithm for operon prediction. *Bioinformatics*, **21**, 1403 – 1407.

Jin, W., Riley, R. M., Wolfinger, R. D., White, K. P., Passador-Gurgel, G. and Gibson, G. (2001). The contributions of sex, genotype and age to transcriptional variance in *Drosophila melanogaster*. *Nat Genet*, **29**, 389 – 395.

Kanehisa, M. and Bork, P. (2003). Bioinformatics in the post-sequence era. *Nat Genet*, **33**, 305 – 310.

Kanehisa, M., Goto, S., Hattori, M., Aoki-Kinoshita, K. F., Itoh, M., Kawashima, S., Katayama, T., Araki, M. and Hirakawa, M. (2006). From genomics to chemical genomics: new developments in kegg. *Nucleic Acids Res*, **34**, D354 – D357.

Kang, Y., Weber, K. D., Qiu, Y., Kiley, P. J. and Blattner, F. R. (2005). Genome-wide expression analysis indicates that FNR of *Escherichia coli* K-12 regulates a large number of genes of unknown function. *J Bacteriol*, **187**, 1135 – 60.

Keijser, B. J. F., Beek, A. T., Rauwerda, H., Schuren, F., Montijn, R., van der Spek, H. and Brul, S. (2007). Analysis of temporal gene expression during *Bacillus subtilis* spore germination and outgrowth. *J Bacteriol*, **189**, 3624 – 3634.

Kerrien, S., Alam-Faruque, Y., Aranda, B., Bancarz, I., Bridge, A., Derow, C., Dimmer, E., Feuermann, M., Friedrichsen, A., Huntley, R., Kohler, C., Khadake, J., Leroy, C., Liban, A., Lieftink, C., Montecchi-Palazzi, L., Orchard, S., Risse, J., Robbe, K., Roechert, B., Thorneycroft, D., Zhang, Y., Apweiler, R. and Hermjakob, H. (2007). IntAct–open source resource for molecular interaction data. *Nucleic Acids Res*, **35**, D561 – D565.

Kersey, P., Bower, L., Morris, L., Horne, A., Petryszak, R., Kanz, C., Kanapin, A., Das, U., Michoud, K., Phan, I., Gattiker, A., Kulikova, T., Faruque, N., Duggan, K., Mclaren, P., Reimholz, B., Duret, L., Penel, S., Reuter, I. and Apweiler, R. (2005). Integr8 and genome reviews: integrated views of complete genomes and proteomes. *Nucleic Acids Res*, **33**, D297 – D302.

Khatri, P. and Draghici, S. (2005). Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics*, **21**, 3587 – 3595.

Kiley, P. J. and Beinert, H. (2003). The role of fe-s proteins in sensing and regulation in bacteria. *Curr Opin Microbiol*, **6**, 181 – 185.

Knudsen, S. (2002). *A Biologist's guide to Analysis of DNA microarray data*. John Willey and Sons, Denmark.

Koike, A. and Takagi, T. (2004). Gene/protein/family name recognition in biomedical literature. In *Proceedings of HLT/NAACL BioLINK Workshop*. 9 – 16.

Kreutzberger, J. (2006). Protein microarrays: a chance to study microorganisms? *Appl Microbiol Biotechnol*, **70**, 383 – 390.

Krieger, R. (2001). *Transkriptionelle Kontrolle der Denitrifikation in Pseudomonas aeruginosa*. Ph.D. thesis, University of Freiburg, Germany.

Krull, M., Pistor, S., Voss, N., Kel, A., Reuter, I., Kronenberg, D., Michael, H., Schwarzer, K., Potapov, A., Choi, C., Kel-Margoulis, O. and Wingender, E. (2006). Transpath: an information resource for storing and visualizing signaling pathways and their pathological aberrations. *Nucleic Acids Res*, **34**, D546 – D551.

Kulikova, T., Akhtar, R., Aldebert, P., Althorpe, N., Andersson, M., Baldwin, A., Bates, K., Bhattacharyya, S., Bower, L., Browne, P., Castro, M., Cochrane, G., Duggan, K., Eberhardt, R., Faruque, N., Hoad, G., Kanz, C., Lee, C., Leinonen, R., Lin, Q., Lombard, V., Lopez, R., Lorenc, D., McWilliam, H., Mukherjee, G., Nardone, F., Pastor, M. P. G., Plaister, S., Sobhany, S., Stoehr, P., Vaughan, R., Wu, D., Zhu, W. and Apweiler, R. (2007). EMBL Nucleotide Sequence Database in 2006. *Nucleic Acids Res*, **35**, D16 – D20.

Köhler, W., Schachtel, G. and Voleske, P. (2002). *Biostatistik (in German).* Springer, Inc.

Lee, M. L., Kuo, F. C., Whitmore, G. A. and Sklar, J. (2000). Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cdna hybridizations. *Proc Natl Acad Sci U S A*, **97**, 9834 – 9839.

Li, C. and Wong, W. H. (2001a). Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection. *Proc Natl Acad Sci U S A*, **98**, 31 – 36.

Li, C. and Wong, W. H. (2001b). Model-based analysis of oligonucleotide arrays: model validation, design issues and standard error application. *Genome Biol*, **2**, RESEARCH0032.

Lipshutz, R. J., Fodor, S. P., Gingeras, T. R. and Lockhart, D. J. (1999). High density synthetic oligonucleotide arrays. *Nat Genet*, **21**, 20 – 24.

Luscombe, N., Greenbaum, D. and Gerstein, M. (2001). What is bioinformatics? an introduction and overview. *Intl Medical Informatics Association (Yearbook)*, 83 – 99.

Madeira, S. and Oliveira, A. (2004). Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **1**, 24 – 45.

Madigan, M. T. and Martinko, J. M. (2006). *Brock - Biology of Microorganisms*. Prentice Hall, 11th edition. ISBN 0-13-196893-9.

Malpica, R., Franco, B., Rodriguez, C., Kwon, O. and Georgellis, D. (2004). Identification of a quinone-sensitive redox switch in the arcb sensor kinase. *Proc Natl Acad Sci U S A*, **101**, 13318 – 13323.

Martin, D., Brun, C., Remy, E., Mouren, P., Thieffry, D. and Jacq, B. (2004). Gotoolbox: functional analysis of gene datasets based on gene ontology. *Genome Biol*, **5**, R101.

Matys, V., Fricke, E., Geffers, R., Gössling, E., Haubrock, M., Hehl, R., Hornischer, K., Karas, D., Kel, A. E., Kel-Margoulis, O. V., Kloos, D.-U., Land, S., Lewicki-Potapov, B., Michael, H., Münch, R., Reuter, I., Rotert, S., Saxel, H., Scheer, M., Thiele, S. and Wingender, E. (2003). Transfac: transcriptional regulation, from patterns to profiles. *Nucleic Acids Res*, **31**, 374 – 378.

Matys, V., Kel-Margoulis, O. V., Fricke, E., Liebich, I., Land, S., Barre-Dirrie, A., Reuter, I., Chekmenev, D., Krull, M., Hornischer, K., Voss, N., Stegmaier, P., Lewicki-Potapov, B., Saxel, H., Kel, A. E. and Wingender, E. (2006). Transfac and its module transcompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Res*, **34**, D108 – D110.

McCarroll, S. A., Murphy, C. T., Zou, S., Pletcher, S. D., Chin, C.-S., Jan, Y. N., Kenyon, C., Bargmann, C. I. and Li, H. (2004). Comparing genomic expression patterns across species identifies shared transcriptional profile in aging. *Nat Genet*, **36**, 197 – 204.

Millenaar, F. F., Okyere, J., May, S. T., van Zanten, M., Voesenek, L. A. C. J. and Peeters, A. J. M. (2006). How to decide? Different methods of calculating gene expression from short oligonucleotide array data will give different results. *BMC Bioinformatics*, **7**, 137.

Mootha, V. K., Lindgren, C. M., Eriksson, K.-F., Subramanian, A., Sihag, S., Lehar, J., Puigserver, P., Carlsson, E., Ridderstrale, M., Laurila, E., Houstis, N., Daly, M. J., Patterson, N., Mesirov, J. P., Golub, T. R., Tamayo, P., Spiegelman, B., Lander, E. S., Hirschhorn, J. N., Altshuler, D. and Groop, L. C. (2003). PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nat Genet*, **34**, 267 – 273.

Morrison, D. A. and Ellis, J. T. (2003). The design and analysis of microarray experiments: applications in parasitology. *DNA Cell Biol*, **22**, 357 – 394.

Münch, R., Hiller, K., Barg, H., Heldt, D., Linz, S., Wingender, E. and Jahn, D. (2003). PRODORIC: prokaryotic database of gene regulation. *Nucleic Acids Res*, **31**, 266 – 269.

Münch, R., Hiller, K., Grote, A., Scheer, M., Klein, J., Schobert, M. and Jahn, D. (2005). Virtual Footprint and PRODORIC: an integrative framework for regulon prediction in prokaryotes. *Bioinformatics*, **21**, 4187 – 4189.

Naef, F., Lim, D. A., Patil, N. and Magnasco, M. (2002). Dna hybridization to mismatched templates: A chip study. *Phys Rev E*, **65**, 040902.

Naef, F., Lim, D. A., Patil, N. and Magnasco, M. O. (2001). From features to expression: High-density oligonucleotide array analysis revisited. In *Proceedings of the DIMACS Workshop on Analysis of Gene Expression Data 2001.*

Naidoo, S., Denbyb, K. J. and Berger, D. K. (2005). Microarray experiments: considerations for experimental design. *S Afr J Sci*, **101**, 347 – 354.

Nam, D. and Kim, S.-Y. (2008). Gene-set approach for expression pattern analysis. *Brief Bioinform*, **9**, 189 – 197.

Navarro, J. D., Niranjan, V., Peri, S., Jonnalagadda, C. K. and Pandey, A. (2003). From biological databases to platforms for biomedical discovery. *Trends Biotechnol*, **21**, 263 – 268.

Nicholson, W. L., Munakata, N., Horneck, G., Melosh, H. J. and Setlow, P. (2000). Resistance of bacillus endospores to extreme terrestrial and extraterrestrial environments. *Microbiol Mol Biol Rev*, **64**, 548 – 572.

Oberg, A. L., Mahoney, D. W., Ballman, K. V. and Therneau, T. M. (2006). Joint estimation of calibration and expression for high-density oligonucleotide arrays. *Bioinformatics*, **22**, 2381 – 2387.

Pan, K.-H., Lih, C.-J. and Cohen, S. N. (2005). Effects of threshold choice on biological conclusions reached during analysis of gene expression by DNA microarrays. *Proc Natl Acad Sci U S A*, **102**, 8961 – 8965.

Parkinson, H., Kapushesky, M., Shojatalab, M., Abeygunawardena, N., Coulson, R., Farne, A., Holloway, E., Kolesnykov, N., Lilja, P., Lukk, M., Mani, R., Rayner, T., Sharma, A., William, E., Sarkans, U. and Brazma, A. (2007). Arrayexpress–a public database of microarray experiments and gene expression profiles. *Nucleic Acids Res*, **35**, D747 – D750.

Pillet, V., Zehnder, M., Seewald, A. K., Veuthey, A.-L. and Petrak, J. (2005). GPSDB: a new database for synonyms expansion of gene and protein names. *Bioinformatics*, **21**, 1743 – 1744.

Preiss, J. and Romeo, T. (1994). Molecular biology and regulatory aspects of glycogen biosynthesis in bacteria. *Prog Nucleic Acid Res Mol Biol*, **47**, 299 – 329.

Price, M. N., Huang, K. H., Alm, E. J. and Arkin, A. P. (2005). A novel method for accurate operon predictions in all sequenced prokaryotes. *Nucleic Acids Res*, **33**, 880 – 892.

Rocke, D. M. and Durbin, B. (2001). A model for measurement error for gene expression arrays. *J Comput Biol*, **8**, 557 – 569.

Salgado, H., Gama-Castro, S., Peralta-Gil, M., Díaz-Peredo, E., Sánchez-Solano, F., Santos-Zavaleta, A., Martínez-Flores, I., Jiménez-Jacinto, V., Bonavides-Martínez, C., Segura-Salazar, J., Martínez-Antonio, A. and Collado-Vides, J. (2006). RegulonDB (version 5.0): *Escherichia coli* K-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Res*, **34**, D394 – D397.

Salmon, K., Hung, S., Mekjian, K., Baldi, P., Hatfield, G. W. and Gunsalus, R. P. (2003). Global gene expression profiling in *Escherichia coli* K12. The effects of oxygen availability and FNR. *J Biol Chem*, **278**, 29837 – 29855.

Salmon, K. A., Hung, S., Steffen, N. R., Krupp, R., Baldi, P., Hatfield, G. W. and Gunsalus, R. P. (2005). Global gene expression profiling in *Escherichia coli* K12: effects of oxygen availability and ArcA. *J Biol Chem*, **280**, 15084 – 15096.

Saviozzi, S. and Calogero, R. A. (2003). Microarray probe expression measures, data normalization and statistical validation. *Comparative and Functional Genomics*, **4**, 442 – 446.

Schadt, E. E., Li, C., Ellis, B. and Wong, W. H. (2001). Feature extraction and normalization algorithms for high-density oligonucleotide gene expression array data. *J Cell Biochem Suppl*, **Suppl 37**, 120 – 125.

Scheer, M., Klawonn, F., Münch, R., Grote, A., Hiller, K., Choi, C., Koch, I., Schobert, M., Härtig, E., Klages, U. and Jahn, D. (2006). JProGO: a novel tool for the functional interpretation of prokaryotic microarray data using Gene Ontology information. *Nucleic Acids Res*, **34**, W510 – W515.

Schena, M. (2003). *Microarray Analysis*. John Wiley & Sons, New York, 1. edition.

Schena, M., Shalon, D., Davis, R. W. and Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, **270**, 467 – 470.

Schomburg, I., Chang, A., Ebeling, C., Gremse, M., Heldt, C., Huhn, G. and Schomburg, D. (2004). Brenda, the enzyme database: updates and major new developments. *Nucleic Acids Res*, **32**, D431 – D433.

Schreiber, K., Krieger, R., Benkert, B., Eschbach, M., Arai, H., Schobert, M. and Jahn, D. (2007). The anaerobic regulatory network required for *Pseudomonas aeruginosa* nitrate respiration. *J Bacteriol*, **189**, 4310 – 4314.

Seo, J. and Hoffman, E. P. (2006). Probe set algorithms: is there a rational best bet? *BMC Bioinformatics*, **7**, 395.

Sharma, V., Noriega, C. E. and Rowe, J. J. (2006). Involvement of nark1 and nark2 proteins in transport of nitrate and nitrite in the denitrifying bacterium pseudomonas aeruginosa pao1. *Appl Environ Microbiol*, **72**, 695 – 701.

Shedden, K., Chen, W., Kuick, R., Ghosh, D., Macdonald, J., Cho, K. R., Giordano, T. J., Gruber, S. B., Fearon, E. R., Taylor, J. M. G. and Hanash, S. (2005). Comparison of seven methods for producing Affymetrix expression scores based on False Discovery Rates in disease profiling data. *BMC Bioinformatics*, **6**, 26.

Shen-Orr, S. S., Milo, R., Mangan, S. and Alon, U. (2002). Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nat Genet*, **31**, 64 – 68.

Singh, O. V. and Nagaraj, N. S. (2006). Transcriptomics, proteomics and interactomics: unique approaches to track the insights of bioremediation. *Brief Funct Genomic Proteomic*, **4**, 355 – 362.

Smid, M. and Dorssers, L. C. J. (2004). GO-Mapper: functional analysis of gene expression data using the expression level as a score to evaluate Gene Ontology terms. *Bioinformatics*, **20**, 2618 – 2625.

Spiro, S. and Guest, J. R. (1991). Adaptive responses to oxygen limitation in escherichia coli. *Trends Biochem Sci*, **16**, 310 – 314.

Steege, D. A. (2000). Emerging features of mRNA decay in bacteria. *RNA*, **6**, 1079 – 1090.

Steinhauser, D., Junker, B. H., Luedemann, A., Selbig, J. and Kopka, J. (2004). Hypothesis-driven approach to predict transcriptional units from gene expression data. *Bioinformatics*, **20**, 1928 – 1939.

Sterk, P., Kersey, P. J. and Apweiler, R. (2006). Genome reviews: standardizing content and representation of information about complete genomes. *OMICS*, **10**, 114 – 118.

Stoughton, R. B. (2005). Applications of DNA microarrays in biology. *Annu Rev Biochem*, **74**, 53 – 82.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S. and Mesirov, J. P. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci U S A*, **102**, 15545 – 15550.

Sugawara, H., Ogasawara, O., Okubo, K., Gojobori, T. and Tateno, Y. (2008). DDBJ with new system and face. *Nucleic Acids Res*, **36**, D22 – D24.

Tao, H., Bausch, C., Richmond, C., Blattner, F. R. and Conway, T. (1999). Functional genomics: expression analysis of escherichia coli growing on minimal and rich media. *J Bacteriol*, **181**, 6425 – 6440.

Tatusov, R. L., Natale, D. A., Garkavtsev, I. V., Tatusova, T. A., Shankavaram, U. T., Rao, B. S., Kiryutin, B., Galperin, M. Y., Fedorova, N. D. and Koonin, E. V. (2001). The cog database: new developments in phylogenetic classification of proteins from complete genomes. *Nucleic Acids Res*, **29**, 22 – 28.

Tusher, V. G., Tibshirani, R. and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proc Natl Acad Sci U S A*, **98**, 5116 – 5121.

Venkatasubbarao, S. (2004). Microarrays–status and prospects. *Trends Biotechnol*, **22**, 630 – 637.

Volinia, S., Evangelisti, R., Francioso, F., Arcelli, D., Carella, M. and Gasparini, P. (2004). Goal: automated gene ontology analysis of expression profiles. *Nucleic Acids Res*, **32**, W492 – W499.

von Mering, C., Jensen, L. J., Kuhn, M., Chaffron, S., Doerks, T., Krüger, B., Snel, B. and Bork, P. (2007). String 7–recent developments in the integration and prediction of protein interactions. *Nucleic Acids Res*, **35**, D358 – D362.

Webb, E. C., of the International Union of Biochemistry, N. C. and Biology, M. (1992). *Enzyme Nomenclature: Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes*. Academic Press, New York, NY.

Westover, B. P., Buhler, J. D., Sonnenburg, J. L. and Gordon, J. I. (2005). Operon prediction without a training set. *Bioinformatics*, **21**, 880 – 888.

Wheeler, D. L., Chappey, C., Lash, A. E., Leipe, D. D., Madden, T. L., Schuler, G. D., Tatusova, T. A. and Rapp, B. A. (2000). Database resources of the national center for biotechnology information. *Nucleic Acids Res*, **28**, 10 – 14.

Wolfe, C. J., Kohane, I. S. and Butte, A. J. (2005). Systematic survey reveals general applicability of "guilt-by-association" within gene coexpression networks. *BMC Bioinformatics*, **6**, 227.

Wolfinger, R. D., Gibson, G., Wolfinger, E. D., Bennett, L., Hamadeh, H., Bushel, P., Afshari, C. and Paules, R. S. (2001). Assessing gene significance from cdna microarray expression data via mixed models. *J Comput Biol*, **8**, 625 – 637.

Wu, C. H., Apweiler, R., Bairoch, A., Natale, D. A., Barker, W. C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M. J., Mazumder, R., O'Donovan, C., Redaschi, N. and Suzek, B. (2006). The Universal Protein Resource (UniProt): an expanding universe of protein information. *Nucleic Acids Res*, **34**, D187 – D191.

Wu, Z. and Irizarry, R. A. (2005). Stochastic models inspired by hybridization theory for short oligonucleotide arrays. *J Comput Biol*, **12**, 882 – 893.

Wu, Z. and Irizarry, R. A. (2007). A statistical framework for the analysis of microarray probe-level data. *Ann Appl Statist*, **1**, 333 – 357.

Xiao, Y., Gordon, A. and Yakovlev1, A. (2006). The l1-version of the cramér-von mises test for two-sample comparisons in microarray data analysis. *EURASIP J Bioinform Syst Biol*, **2006**, 85769.

Yang, J., Buckley, M., Dudoit, S. and Speed, T. (2002). Comparison of methods for image analysis on cDNA microarray data. *Journal of Computational and Graphical Statistics*, **11**, 108 – 136.

Yauk, C. L., Berndt, M. L., Williams, A. and Douglas, G. R. (2004). Comprehensive comparison of six microarray technologies. *Nucleic Acids Res*, **32**, e124.

Ye, R. W., Tao, W., Bedzyk, L., Young, T., Chen, M. and Li, L. (2000). Global gene expression profiles of *Bacillus subtilis* grown under anaerobic conditions. *J Bacteriol*, **182**, 4458 – 4465.

Ye, R. W., Wang, T., Bedzyk, L. and Croker, K. M. (2001). Applications of DNA microarrays in microbial systems. *J Microbiol Methods*, **47**, 257 – 272.

Zareparsi, S., Hero, A., Zack, D. J., Williams, R. W. and Swaroop, A. (2004). Seeing the unseen: Microarray-based gene expression profiling in vision. *Invest Ophthalmol Vis Sci*, **45**, 2457 – 2462.

Zeeberg, B., Qin, H., Narasimhan, S., Sunshine, M., Cao, H., Kane, D., Reimers, M., Stephens, R., Bryant, D., Burt, S., Elnekave, E., Hari, D., Wynn, T., Cunningham-Rundles, C., Stewart, D., Nelson, D. and Weinstein, J. (2005). High-Throughput GoMiner, an 'industrial-strength' integrative Gene Ontology tool for interpretation of multiple-microarray experiments, with application to studies of Common Variable Immune Deficiency (CVID). *BMC Bioinformatics*, **6**, 168.

Zeeberg, B. R., Feng, W., Wang, G., Wang, M. D., Fojo, A. T., Sunshine, M., Narasimhan, S., Kane, D. W., Reinhold, W. C., Lababidi, S., Bussey, K. J., Riss, J., Barrett, J. C. and Weinstein, J. N. (2003). GoMiner: a resource for biological interpretation of genomic and proteomic data. *Genome Biol*, **4**, R28.

Zhang, B., Schmoyer, D., Kirov, S. and Snoddy, J. (2004). GOTree machine (GOTM): a web-based platform for interpreting sets of interesting genes using gene ontology hierarchies. *BMC Bioinformatics*, **5**, 16.

Zhang, Q., Ushijima, R., Kawai, T. and Tanaka, H. (2005). Which to use? - microarray data analysis in input and output data processing. *Chem Bio Inform J*, **4**, 56 – 72.

Zhong, S., Storch, K.-F., Lipan, O., Kao, M.-C. J., Weitz, C. J. and Wong, W. H. (2004). GoSurfer: a graphical interactive tool for comparative analysis of large gene sets in Gene Ontology space. *Appl Bioinformatics*, **3**, 261 – 264.

Zöfel, P. (2002). *Statistik verstehen. (in German).* Addison Wesley, Munich.

Appendices

SQL statement on the polypeptide2class table: to obtain 4^{th} level GO nodes of MF

INSERT INTO polypeptide2class (class_no, polypeptide_no) (SELECT DISTINCT class.class_no, go2polypeptide.polypeptide_no FROM class, go, graph_path, go2polypeptide WHERE go.go_no=graph_path.go1_no AND graph_path.go2_no = go2polypeptide.go_no AND go.go_acc = class.class_acc AND go.go_no IN (SELECT go_no FROM go2go WHERE parent_go_no IN ((SELECT go_no FROM go WHERE go_name='molecular_function'))) ORDER BY class.class_no, go2polypeptide.polypeptide_no)

Further figures

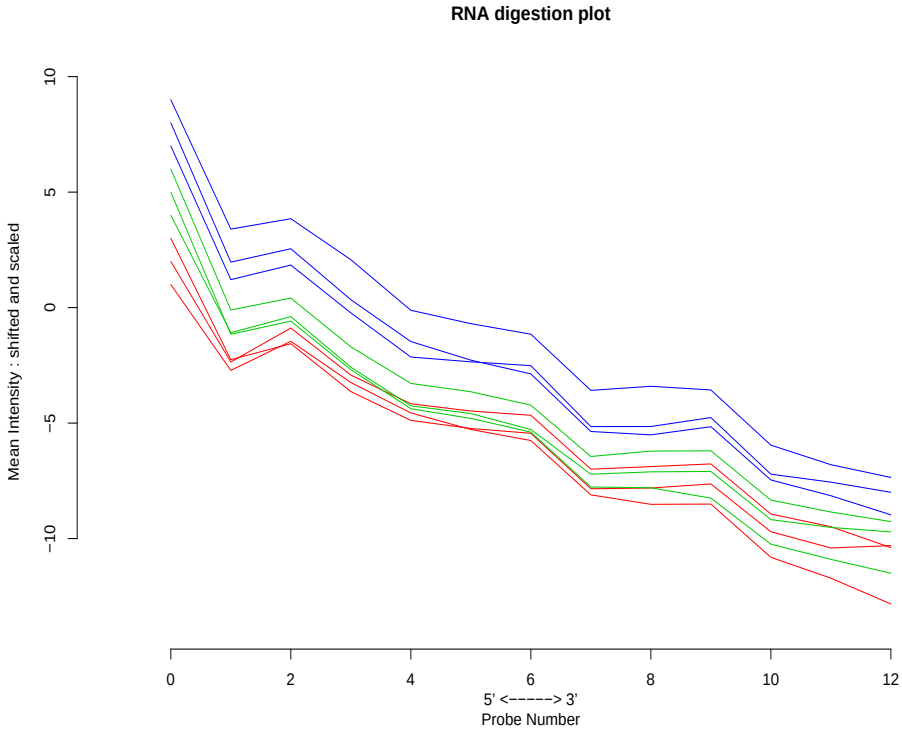
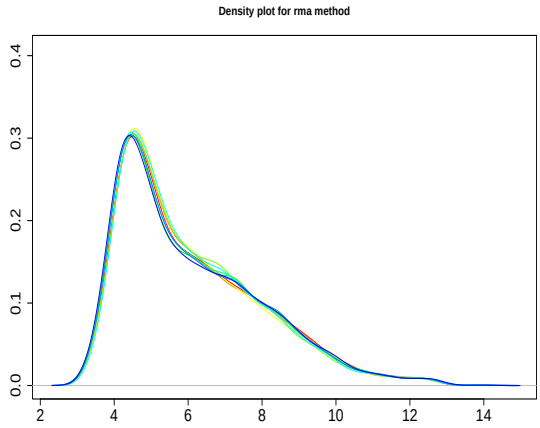
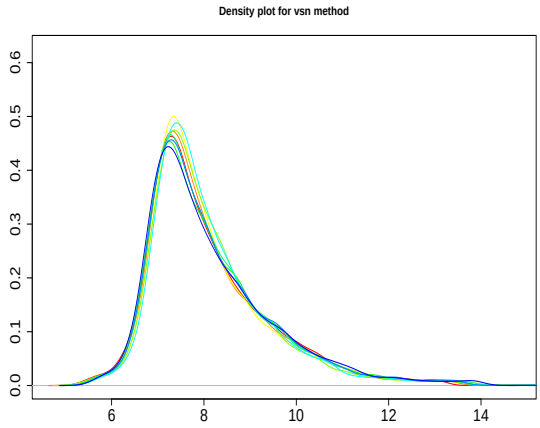


Figure 30: RNA degradation plot of the raw data of Benkert et al., 2008 (unpubl. res.). Red lines represent the three replicates of PAO1 wild type cells grown anaerobically with nitrate, the green lines the wild types cells and the blue lines the *narL* mutant cells, both grown without nitrate

a)



b)



c)

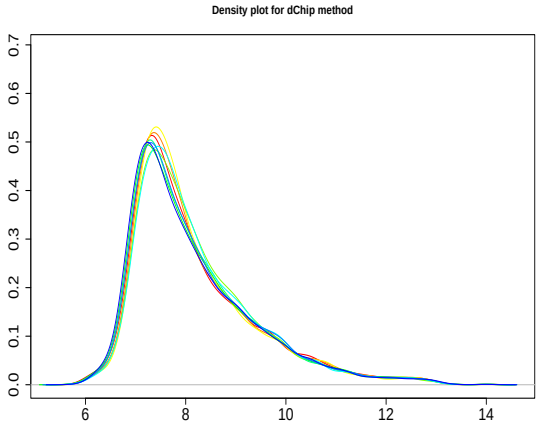


Figure 31: Density plots of the preprocessed data of Benkert et al., 2008 (unpubl. res.). The order of subfigures and legends are the same as in Fig. 24 (see for there details).

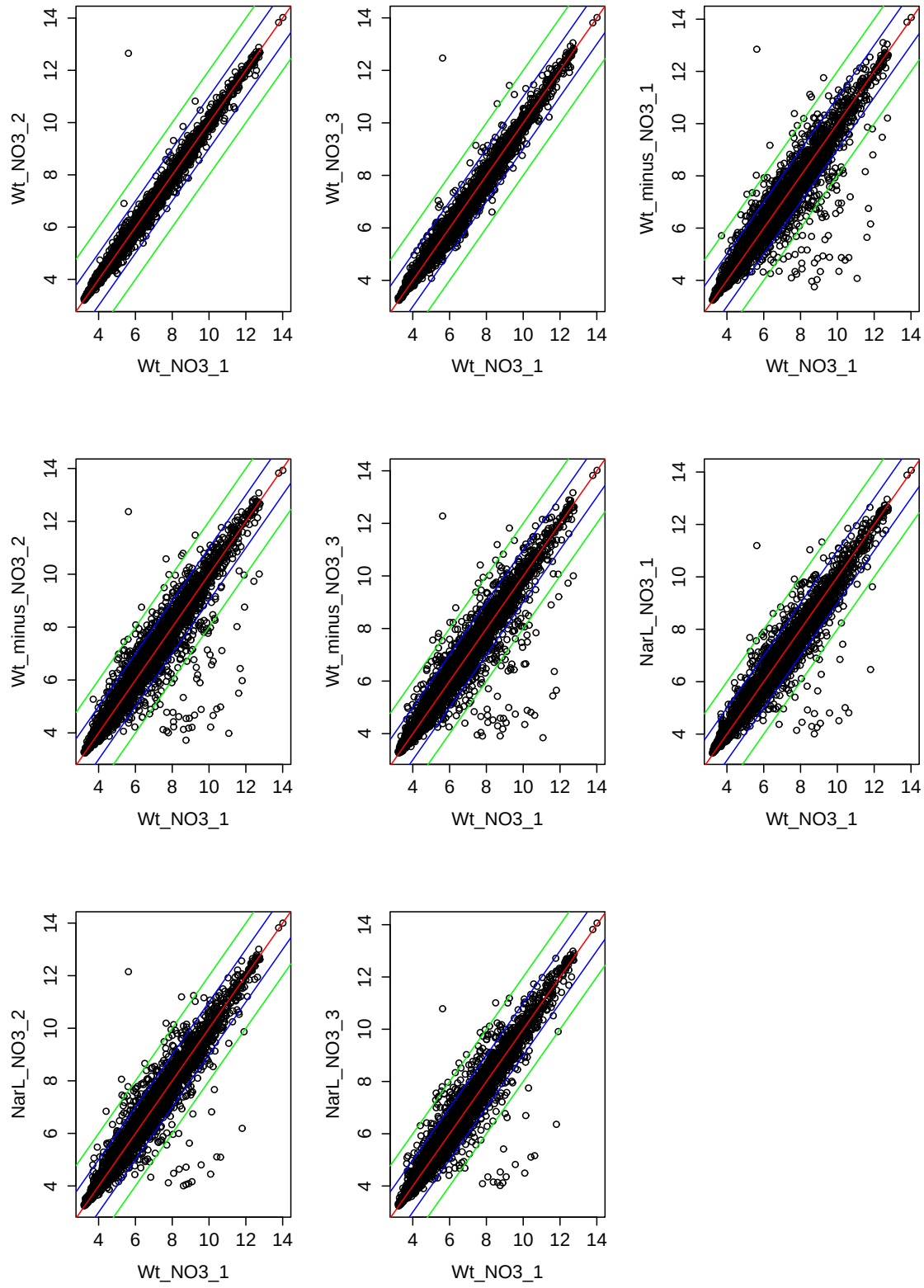


Figure 32: Scatter plots of the 9 preprocessed data sets of Benkert et al., 2008 (unpubl. res.). One array, wild type grown aerobically with nitrate (Wt_NO3_1), was selected and compared to the other eight arrays (Wt_NO3, Wt_minus_NO3, NarL_NO3). The two replicates (Wt_NO3_2 and Wt_NO3_3) show the highest agreement. The red line represents the diagonal, the blue lines the diagonal shifted by one log step and the green lines shifted by two log steps.

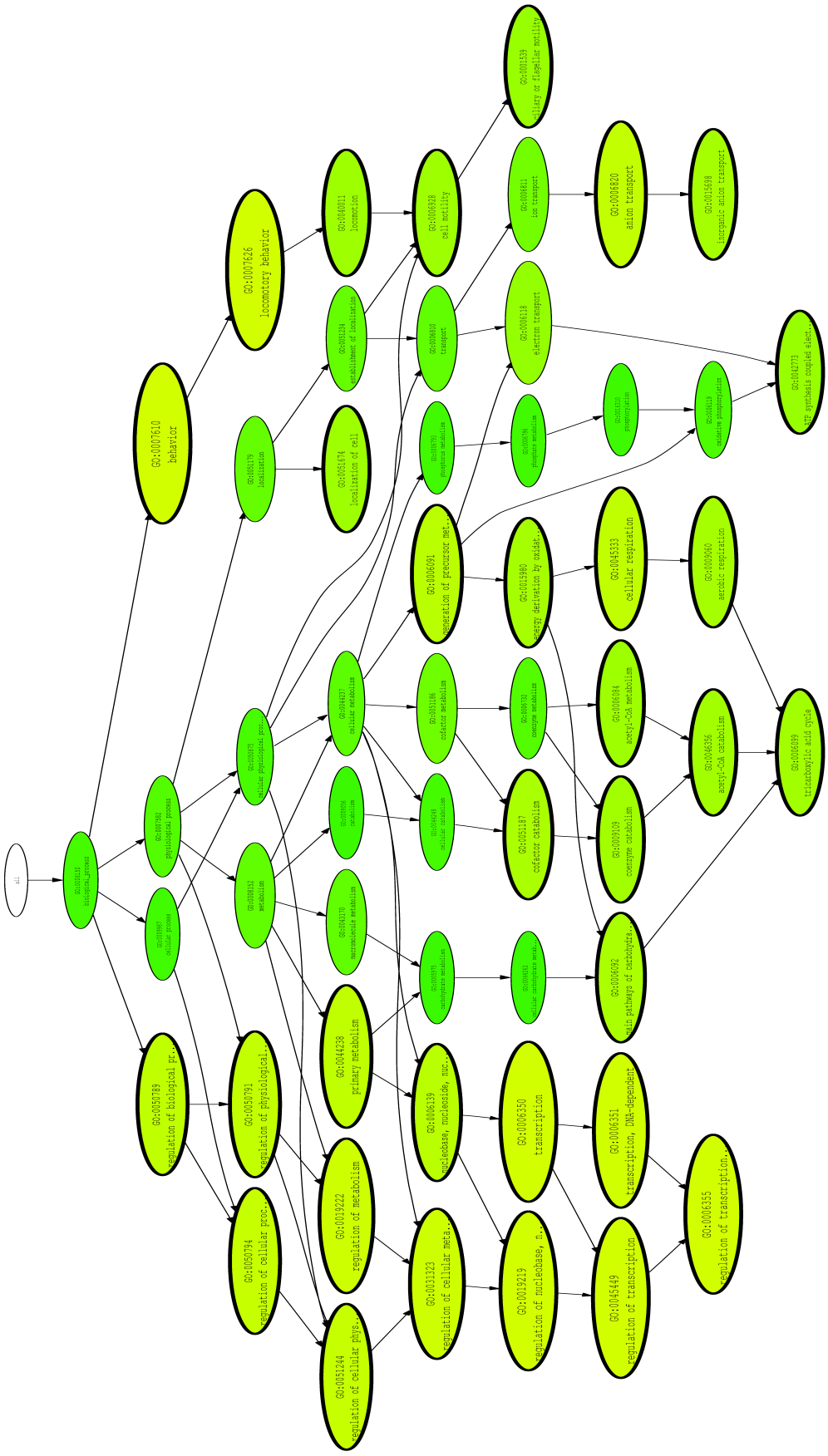


Figure 33: *Biological Process* subgraph of the JProGO results for the expression data representing the PAO1 wild type cells grown anaerobically with versus without nitrate (Benkert et al., 2008, unpubl. res). The ppde computed for the rma-preprocessed data were used as input data for JProGO. A two-sided Mann-Whitney U-test was used with an FDR of 10%. All significant nodes and the paths up to the root node ('*all*') are present..

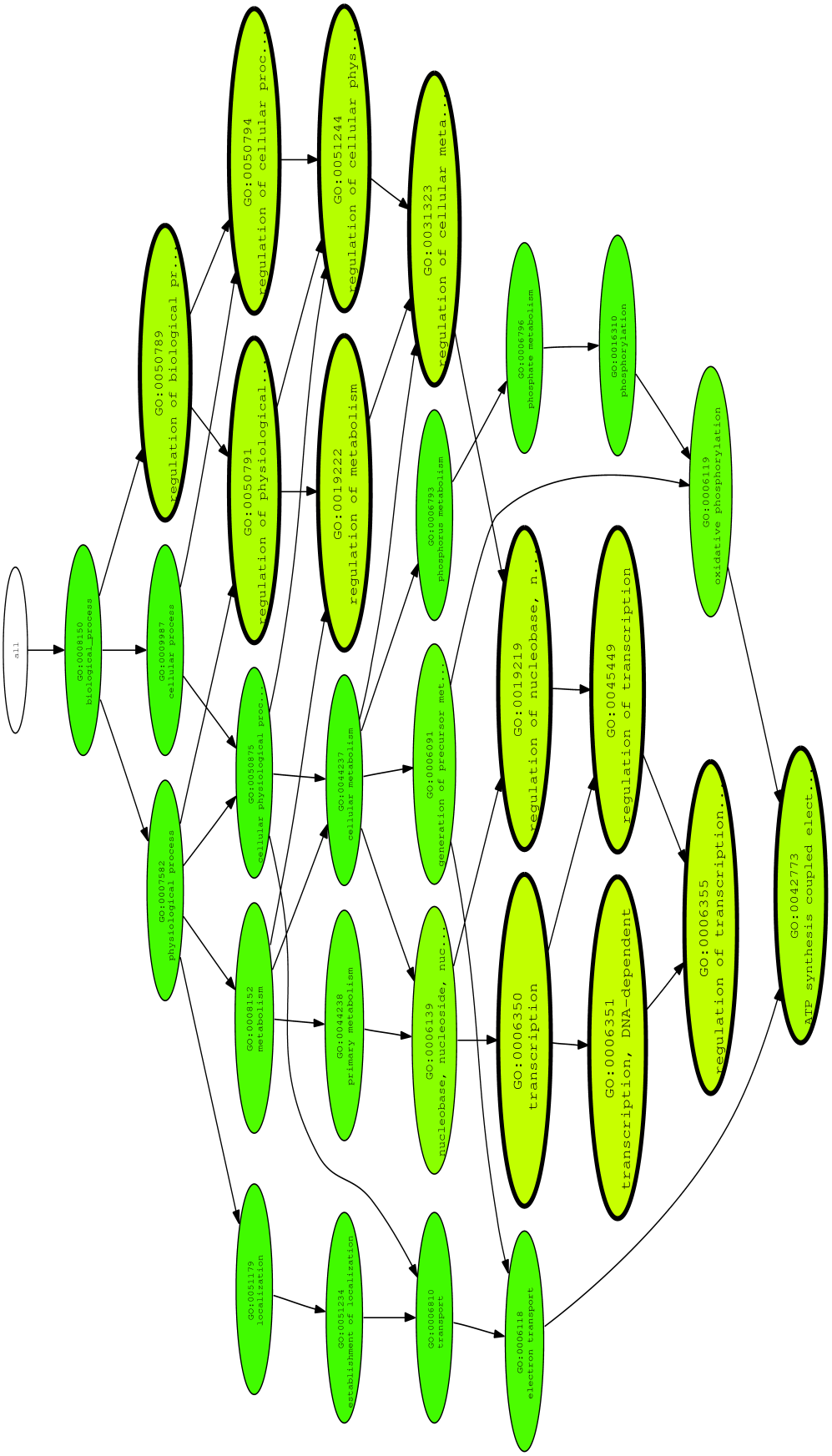


Figure 34: *Biological Process* subgraph of the JProGO results for the expression data representing the PAO1 *narL* strain versus wild type cells grown anaerobically, both with nitrate (Benkert et al., 2008, unpubl. res). The ppde computed for the rma-preprocessed data were used as input data for JProGO. A two-sided Mann-Whitney U-test was used with an FDR of 10%. All significant nodes and the paths up to the root node ('*all*') are present.

Table 20: *E. coli* regulons ranked by the p-values obtained for the different microarray data sets (pde) using the Kolmogorov-Smirnov test of JRegA. The organization is exactly the same as that of Table 18, the only difference between both tables is the used statistical test: KS-Test instead of the Mann-Whitney U-test.

Regulon	∑Genes	$arcA^-$/Wt -O_2 Salmon,2005	fnr^-/Wt -O_2 Salmon,2003	fnr^-/Wt -O_2 Kang,2005	fnr^- +O_2/-O_2 Kang,2005	fnr^-/Wt +O_2 Kang,2005	lrp^- Wt Hung,2002	Wt +O_2/-O_2 Salmon,2003	Wt +O_2/-O_2 Kang,2005
araC	10	22	33	32	33	17	21	12	32
arcA	69	17	24	4	1	3	37	2	1
argR	17	8	13	34	10	7	3	9	41
caiF	10	18	6	9	34	41	20	24	18
cbl	9	21	35	20	23	12	35	27	23
cpxR	28	19	32	7	5	33	38	6	34
crp	308	2	2	16	16	6	5	3	9
cysB	17	5	41	21	13	5	33	16	37
cytR	10	16	21	28	39	36	19	31	20
dnaA	15	25	18	41	38	37	29	17	30
fadR	9	40	38	19	7	28	13	40	40
fhlA	17	6	16	37	22	16	6	33	6
fis	131	14	8	31	30	10	34	4	14
fliA	32	33	36	1	2	14	25	39	19
fnr	62	7	1	2	21	2	8	14	3
fruR	13	11	20	22	11	23	41	23	11
fur	26	4	15	13	14	11	27	35	5
glnG	11	37	23	25	24	34	15	32	22
glpR	8	23	34	8	31	32	9	10	7
gntR	9	10	25	26	18	27	26	13	28
lexA	87	28	14	24	36	21	32	41	36
lrp	22	15	10	5	9	22	2	36	31
malT	9	29	39	39	19	8	1	5	39
marA	25	9	31	15	12	26	11	34	16
metJ	35	35	22	18	26	13	16	37	25
modE	26	39	9	12	29	30	30	25	13
nagC	10	20	17	38	40	40	31	18	33
narL	75	12	12	6	8	1	23	7	2
narP	14	24	3	36	3	4	24	26	4
ompR	9	31	30	10	6	20	14	29	35
oxyR	25	1	4	17	32	31	39	22	8
phoB	30	13	19	23	28	25	12	8	29
phoP	10	32	11	30	41	38	22	20	17
purR	46	36	28	27	17	24	10	11	38
rob	19	38	40	40	37	9	36	15	21
rpoN	30	27	27	33	27	19	7	38	10
soxS	38	34	37	11	15	15	40	28	15
trpR	9	30	29	29	35	35	28	19	24
tyrR	8	41	26	35	25	29	18	21	26
yhiX	14	3	7	3	4	18	4	1	12
yiaJ	9	26	5	14	20	39	17	30	27

Table 21: *E. coli* regulons ranked by the p-values obtained for different microarray data (pde) sets using the t-test of JRegA. The organization is exactly the same as that of Table 18, the only difference between both tables is the used statistical test: t-test instead of the Mann-Whitney U-test.

Regulon	∑Genes	$arcA^-$/Wt -O_2 Salmon,2005	fnr^-/Wt -O_2 Salmon,2003	fnr^-/Wt -O_2 Kang,2005	fnr^- +O_2/-O_2 Kang,2005	fnr^-/Wt +O_2 Kang,2005	lrp^- Wt Hung,2002	Wt +O_2/-O_2 Salmon,2003	Wt +O_2/-O_2 Kang,2005
araC	10	6	27	29	24	25	12	4	24
arcA	69	29	19	15	2	4	34	5	1
argR	17	2	5	36	9	9	2	12	39
caiF	10	40	1	12	30	41	21	20	28
cbl	9	41	23	27	19	16	41	10	22
cpxR	28	15	40	8	7	29	33	28	36
crp	308	3	6	40	34	23	4	6	18
cysB	17	19	38	34	10	7	27	26	26
cytR	10	4	21	38	28	39	18	19	20
dnaA	15	21	16	39	29	32	22	21	32
fadR	9	33	29	7	5	24	13	37	40
fhlA	17	17	10	31	23	5	6	40	7
fis	131	12	7	28	31	26	38	1	15
fliA	32	38	28	1	3	12	16	39	16
fnr	62	7	4	5	38	2	7	27	6
fruR	13	14	20	18	8	11	37	9	19
fur	26	13	18	13	11	17	24	32	5
glnG	11	30	31	25	27	30	28	14	21
glpR	8	25	26	10	40	31	8	34	4
gntR	9	28	11	22	22	38	29	3	25
lexA	87	35	9	23	39	20	31	13	29
lrp	22	20	12	3	6	37	9	31	27
malT	9	18	39	32	13	13	1	7	38
marA	25	32	24	14	41	15	19	33	11
metJ	35	36	33	20	26	18	15	41	41
modE	26	26	32	9	20	22	32	24	12
nagC	10	8	14	30	35	40	23	18	35
narL	75	16	37	11	16	1	20	11	2
narP	14	9	8	37	1	3	25	35	3
ompR	9	5	36	4	14	34	17	30	31
oxyR	25	1	3	17	36	35	40	15	10
phoB	30	24	41	21	25	19	26	2	33
phoP	10	22	15	24	37	36	14	23	14
purR	46	31	22	19	21	21	10	16	30
rob	19	37	35	35	32	6	35	38	17
rpoN	30	34	25	41	15	10	5	36	8
soxS	38	39	34	6	17	8	39	25	13
trpR	9	23	30	33	33	28	36	22	34
tyrR	8	27	13	26	12	27	11	17	37
yhiX	14	10	17	2	4	14	3	8	9
yiaJ	9	11	2	16	18	33	30	29	23

Table 22: *E. coli* regulons ranked by the p-values obtained for the expression ratios for different microarray data sets using the KS-test of JRegA. The organization is exactly the same as that of Table 18.

Regulon	∑Genes	$arcA^-$/Wt -O_2 Salmon,2005	fnr^-/Wt -O_2 Salmon,2003	fnr^-/Wt -O_2 Kang,2005	fnr^- +O_2/-O_2 Kang,2005	fnr^-/Wt +O_2 Kang,2005	lrp^- Wt Hung,2002	Wt +O_2/-O_2 Salmon,2003	Wt +O_2/-O_2 Kang,2005
araC	10	35	35	25	19	39	8	14	34
arcA	69	2	2	5	2	1	30	2	3
argR	17	7	5	20	20	21	14	20	12
caiF	10	25	15	17	25	33	15	16	38
cbl	9	34	34	40	35	35	37	40	41
cpxR	28	24	39	22	21	40	21	27	29
crp	308	6	20	12	7	38	1	32	17
cysB	17	40	23	33	32	34	16	15	32
cytR	10	27	30	39	37	32	35	22	33
dnaA	15	12	21	41	28	37	28	13	28
fadR	9	36	40	35	12	24	31	41	39
fhlA	17	17	10	7	36	9	7	21	5
fis	131	1	1	11	5	36	36	1	14
fliA	32	33	32	1	1	11	25	24	9
fnr	62	16	14	3	11	7	6	31	1
fruR	13	4	12	36	31	28	40	6	27
fur	26	10	13	23	6	30	12	11	8
glnG	11	39	37	38	34	3	23	39	11
glpR	8	29	24	28	26	17	13	29	20
gntR	9	14	19	19	39	13	39	19	25
lexA	87	19	4	30	27	22	41	10	26
lrp	22	11	16	9	9	16	19	35	22
malT	9	22	27	29	17	14	2	7	24
marA	25	28	25	10	29	29	22	38	30
metJ	35	41	11	21	33	31	33	36	19
modE	26	32	41	6	13	18	34	23	16
nagC	10	18	31	34	30	27	24	18	35
narL	75	31	18	4	8	2	9	5	2
narP	14	21	9	24	4	5	18	28	4
ompR	9	38	33	14	10	6	38	37	40
oxyR	25	15	22	15	40	15	26	26	7
phoB	30	9	17	32	14	23	5	4	23
phoP	10	13	8	13	41	26	20	25	15
purR	46	23	26	31	15	20	27	9	36
rob	19	30	36	18	23	19	32	34	18
rpoN	30	37	38	16	22	12	4	17	13
soxS	38	5	6	8	24	8	10	30	10
trpR	9	8	3	37	16	41	17	8	31
tyrR	8	20	29	26	18	25	29	33	37
yhiX	14	3	28	2	3	4	3	3	6
yiaJ	9	26	7	27	38	10	11	12	21

Table 23: *E. coli* regulons ranked by the p-values obtained for the expression ratios for different microarray data sets using the t-test of JRegA. The organization is exactly the same as that of Table 18.

Regulon	∑Genes	*arcA*$^-$/Wt -O_2 Salmon,2005	*fnr*$^-$/Wt -O_2 Salmon,2003	*fnr*$^-$/Wt -O_2 Kang,2005	*fnr*$^-$ +O_2/-O_2 Kang,2005	*fnr*$^-$/Wt +O_2 Kang,2005	*lrp*$^-$ Wt Hung,2002	Wt +O_2/-O_2 Salmon,2003	Wt +O_2/-O_2 Kang,2005
araC	10	26	13	10	9	30	5	36	40
arcA	69	12	11	16	21	1	34	41	36
argR	17	1	2	39	5	15	33	7	2
caiF	10	40	17	6	37	27	19	29	34
cbl	9	41	19	25	25	31	41	39	32
cpxR	28	4	14	27	34	38	23	14	37
crp	308	21	28	12	30	29	4	17	41
cysB	17	13	7	20	12	35	30	35	18
cytR	10	31	27	40	36	39	11	32	26
dnaA	15	37	24	32	26	33	21	5	35
fadR	9	29	33	41	11	37	37	38	27
fhlA	17	6	1	4	35	17	7	25	11
fis	131	11	6	30	3	26	32	18	25
fliA	32	9	5	1	1	4	8	31	23
fnr	62	16	39	34	32	23	2	30	3
fruR	13	19	18	31	40	18	31	1	29
fur	26	38	41	38	2	34	17	24	4
glnG	11	28	15	36	31	3	12	16	6
glpR	8	10	10	35	14	11	9	12	31
gntR	9	17	12	5	16	8	36	34	15
lexA	87	3	4	33	39	41	38	27	14
lrp	22	34	31	23	41	7	28	13	7
malT	9	39	36	37	13	28	1	4	12
marA	25	36	23	13	24	25	15	21	38
metJ	35	20	30	7	17	16	39	37	28
modE	26	32	29	2	6	22	14	23	24
nagC	10	25	22	29	38	13	10	9	16
narL	75	24	26	3	23	2	3	40	5
narP	14	7	40	28	19	10	6	28	9
ompR	9	33	34	22	27	5	40	20	30
oxyR	25	35	32	18	18	9	24	11	10
phoB	30	8	8	24	8	40	29	15	13
phoP	10	27	20	8	22	36	27	3	19
purR	46	14	38	9	20	21	25	26	39
rob	19	30	25	26	10	14	35	6	8
rpoN	30	18	37	17	28	19	20	19	20
soxS	38	22	16	21	29	6	16	8	1
trpR	9	23	21	19	4	32	22	2	17
tyrR	8	5	9	15	7	20	26	33	21
yhiX	14	2	35	14	15	12	18	10	22
yiaJ	9	15	3	11	33	24	13	22	33

Table 24: *E. coli* regulons ranked by their p-values obtained for different microarray data sets (pde) using the U-test of JRegA. This Table shows the p-values computed by JRegA which were converted to ranks in Table 18 (see that Table for more information).

Regulon	∑Genes	$arcA^-$/Wt -O_2 Salmon,2005	fnr^-/Wt -O_2 Salmon,2003	fnr^-/Wt -O_2 Kang,2005	fnr^- +O_2/-O_2 Kang,2005	fnr^-/Wt +O_2 Kang,2005	lrp^- Wt Hung,2002	Wt +O_2/-O_2 Salmon,2003	Wt +O_2/-O_2 Kang,2005
crp	308	2,49E-003	1,50E-002	2,61E-001	2,39E-001	2,64E-001	4,46E-004	1,40E-004	5,85E-002
fis	131	1,97E-001	5,67E-002	1,70E-001	4,01E-001	2,91E-001	8,37E-001	3,42E-004	3,36E-002
lexA	87	4,29E-001	1,31E-001	1,57E-001	6,50E-001	3,36E-001	3,75E-001	—	4,42E-001
narL	75	1,82E-001	8,71E-001	1,04E-003	1,18E-002	6,83E-004	1,01E-001	6,62E-003	6,26E-015
arcA	69	6,06E-001	6,28E-001	4,27E-004	4,46E-021	1,04E-001	4,72E-001	9,84E-006	3,19E-019
fnr	62	1,13E-002	3,48E-004	9,13E-007	6,94E-001	6,06E-004	7,95E-003	1,38E-001	7,52E-008
purR	46	9,96E-001	5,12E-001	1,04E-001	2,61E-001	2,75E-001	1,43E-002	1,78E-001	5,64E-001
soxS	38	5,30E-001	9,71E-001	1,13E-003	5,80E-002	7,22E-002	9,19E-001	4,33E-001	2,85E-003
metJ	35	1,00E+000	2,49E-001	6,89E-002	2,75E-001	1,32E-001	6,30E-002	8,69E-001	7,33E-001
fliA	32	8,23E-001	8,60E-001	1,46E-015	3,29E-012	1,10E-001	1,50E-001	9,27E-001	1,27E-001
phoB	30	1,94E-001	4,95E-001	8,10E-002	4,08E-001	3,33E-001	1,66E-001	1,69E-002	6,70E-001
rpoN	30	6,87E-001	5,85E-001	7,19E-001	4,89E-001	7,53E-002	1,18E-002	7,48E-001	3,50E-004
cpxR	28	3,33E-001	5,70E-001	1,45E-003	1,32E-004	4,45E-001	4,89E-001	1,27E-001	9,96E-001
modE	26	9,44E-001	2,87E-001	2,16E-002	2,72E-001	2,69E-001	4,31E-001	1,82E-001	6,45E-003
fur	26	7,85E-003	7,76E-002	1,20E-002	4,85E-002	2,29E-001	1,62E-001	9,66E-001	4,14E-006
marA	25	3,34E-001	4,32E-001	1,12E-002	4,66E-001	1,73E-001	5,83E-002	4,45E-001	2,20E-003
oxyR	25	4,36E-003	9,51E-003	4,17E-002	6,81E-001	9,09E-001	8,77E-001	9,16E-001	1,34E-003
lrp	22	4,02E-001	7,48E-002	7,83E-005	6,53E-004	9,47E-001	1,50E-003	8,43E-001	4,40E-001
rob	19	8,78E-001	7,54E-001	5,38E-001	6,72E-001	2,60E-002	4,52E-001	4,47E-001	9,06E-002
cysB	17	1,44E-001	9,76E-001	3,39E-001	1,33E-002	2,85E-002	2,67E-001	4,26E-001	3,58E-001
argR	17	1,97E-001	8,10E-002	4,72E-001	8,99E-003	9,24E-002	1,81E-002	4,64E-002	9,07E-001
fhlA	17	7,45E-002	2,35E-001	3,28E-001	9,33E-001	3,04E-002	2,77E-002	5,37E-001	5,67E-004
dnaA	15	9,91E-001	2,36E-001	9,76E-001	6,45E-001	7,60E-001	1,68E-001	1,67E-001	8,24E-001
narP	14	3,89E-001	1,16E-002	7,96E-001	2,10E-008	3,64E-002	1,59E-001	7,72E-001	1,04E-005
yhiX	14	1,56E-002	4,29E-002	1,11E-007	8,19E-008	1,18E-001	1,33E-003	9,40E-005	5,41E-004
fruR	13	1,05E-001	1,45E-001	5,13E-002	6,85E-003	3,95E-001	9,12E-001	8,28E-002	3,51E-002
glnG	11	9,43E-001	6,73E-001	1,64E-001	3,14E-001	7,92E-001	2,15E-001	5,89E-001	4,08E-001
phoP	10	4,15E-001	1,47E-001	1,81E-001	8,36E-001	8,10E-001	6,64E-002	1,25E-001	5,44E-002
caiF	10	1,61E-001	2,59E-002	4,60E-003	9,53E-001	9,51E-001	9,88E-002	1,69E-001	3,78E-001
cytR	10	4,02E-001	8,04E-001	4,93E-001	6,69E-001	9,79E-001	1,09E-001	3,59E-001	1,75E-001
araC	10	6,75E-001	5,43E-001	2,19E-001	2,98E-001	2,92E-001	6,41E-002	8,98E-002	3,55E-001
nagC	10	3,43E-001	2,13E-001	4,33E-001	6,71E-001	9,42E-001	1,48E-001	6,23E-001	9,99E-001
yiaJ	9	2,53E-001	1,69E-001	2,98E-002	1,15E-001	8,17E-001	2,57E-001	3,25E-001	2,48E-001
malT	9	4,49E-001	9,57E-001	4,66E-001	9,17E-002	1,72E-001	1,92E-004	7,52E-003	7,89E-001
trpR	9	8,19E-001	4,25E-001	9,04E-001	4,62E-001	5,29E-001	7,92E-001	2,30E-001	6,09E-001
ompR	9	9,66E-001	6,59E-001	1,33E-003	3,83E-003	5,86E-001	4,87E-002	4,18E-001	4,84E-001
cbl	9	2,40E-001	4,49E-001	1,50E-001	1,86E-001	1,19E-001	5,27E-001	3,39E-001	2,28E-001
fadR	9	7,58E-001	6,73E-001	4,75E-002	3,66E-003	3,79E-001	4,15E-002	8,11E-001	9,71E-001
gntR	9	2,41E-001	1,83E-001	8,82E-002	4,35E-001	5,33E-001	2,27E-001	1,35E-001	4,26E-001
glpR	8	6,10E-001	5,43E-001	9,35E-004	5,59E-001	5,72E-001	4,76E-002	3,25E-001	1,07E-004
tyrR	8	9,00E-001	4,57E-001	2,90E-001	5,75E-001	4,13E-001	3,60E-002	1,96E-001	8,02E-001

Table 25: *E. coli* regulons ranked by their p-values obtained for the expression ratios of different microarray data sets using the U-test of JRegA. This Table shows the p-values computed by JRegA which were converted to ranks in Table 19 (see that Table for more information).

Regulon	∑Genes	$arcA^-$/Wt -O_2 Salmon,2005	fnr^-/Wt -O_2 Salmon,2003	fnr^-/Wt -O_2 Kang,2005	fnr^- +O_2/-O_2 Kang,2005	fnr^-/Wt +O_2 Kang,2005	lrp^- Wt Hung,2002	Wt +O_2/-O_2 Salmon,2003	Wt +O_2/-O_2 Kang,2005
araC	10	7,57E-001	2,43E-001	9,08E-002	1,10E-001	4,51E-001	1,74E-003	1,75E-001	3,63E-001
arcA	69	1,15E-006	2,16E-004	3,38E-005	2,31E-003	5,21E-006	2,12E-001	1,23E-006	6,35E-005
argR	17	6,47E-002	3,04E-002	3,04E-002	7,13E-001	6,23E-001	1,33E-001	5,82E-001	1,50E-003
caiF	10	2,70E-001	5,92E-002	8,57E-002	5,85E-001	6,44E-001	2,19E-002	6,00E-002	6,33E-001
cbl	9	7,50E-001	2,65E-001	7,17E-001	5,86E-001	7,34E-001	5,38E-001	9,60E-001	8,93E-001
cpxR	28	4,63E-001	6,65E-001	1,79E-001	8,50E-001	9,22E-001	6,79E-002	8,19E-001	8,66E-002
crp	308	3,63E-001	7,16E-001	2,86E-001	6,02E-002	7,50E-001	1,68E-006	8,93E-001	6,41E-001
cysB	17	9,89E-001	1,47E-001	7,70E-001	9,38E-001	9,20E-001	7,17E-002	2,07E-001	4,49E-001
cytR	10	8,82E-001	7,98E-001	4,00E-001	4,32E-001	8,92E-001	3,10E-001	4,22E-001	7,65E-001
dnaA	15	3,79E-002	4,00E-001	7,83E-001	4,20E-001	9,20E-001	9,94E-001	1,64E-001	2,90E-001
fadR	9	5,72E-001	7,50E-001	6,97E-001	1,11E-002	3,23E-001	8,85E-001	7,68E-001	8,28E-001
fhlA	17	2,31E-002	5,31E-002	1,03E-006	3,52E-001	5,84E-002	3,74E-003	8,55E-002	1,09E-008
fis	131	3,45E-008	6,71E-007	6,36E-002	9,32E-008	8,88E-001	7,07E-001	2,34E-006	1,52E-002
fliA	32	3,77E-001	1,40E-001	3,72E-020	6,87E-019	4,00E-001	5,42E-001	1,34E-001	7,50E-005
fnr	62	1,42E-001	3,04E-001	1,45E-008	1,46E-002	1,00E-001	2,17E-003	4,49E-001	4,91E-012
fruR	13	2,38E-002	5,45E-002	4,29E-001	2,90E-001	5,74E-001	9,86E-001	1,73E-003	1,49E-001
fur	26	4,61E-002	1,84E-001	1,18E-001	1,60E-006	8,95E-001	3,38E-002	4,62E-001	1,84E-006
glnG	11	8,44E-001	3,06E-001	6,19E-001	6,11E-001	4,42E-004	1,01E-001	9,18E-001	5,42E-005
glpR	8	2,86E-001	2,32E-001	4,83E-001	1,04E-001	1,88E-001	2,25E-002	7,38E-001	4,42E-001
gntR	9	1,89E-001	1,33E-001	8,30E-002	7,12E-001	1,45E-002	4,40E-001	2,39E-001	5,81E-002
lexA	87	4,29E-002	1,21E-002	6,15E-001	3,47E-001	3,27E-001	7,27E-001	3,38E-002	2,87E-001
lrp	22	4,84E-002	6,47E-001	1,75E-004	2,26E-002	2,77E-001	3,53E-001	9,21E-001	3,21E-002
malT	9	3,98E-001	4,29E-001	1,76E-001	3,05E-002	2,10E-001	9,11E-004	2,29E-003	9,81E-003
marA	25	5,56E-001	2,41E-001	4,72E-005	6,10E-001	6,47E-001	3,14E-002	8,91E-001	1,37E-001
metJ	35	8,98E-001	4,56E-001	2,62E-002	9,98E-001	3,85E-001	7,03E-001	4,24E-001	1,39E-002
modE	26	4,86E-001	7,33E-001	3,73E-008	4,89E-002	4,26E-001	3,40E-001	6,12E-001	1,32E-003
nagC	10	2,41E-001	2,92E-001	2,95E-001	3,10E-001	3,91E-001	1,54E-001	1,12E-001	2,55E-001
narL	75	5,11E-001	2,24E-001	1,22E-010	6,61E-002	1,29E-004	1,76E-002	2,94E-003	3,31E-010
narP	14	1,93E-001	6,44E-002	4,87E-002	2,28E-009	2,18E-003	8,22E-002	5,02E-001	1,02E-008
ompR	9	6,58E-001	2,25E-001	2,93E-002	1,19E-001	1,33E-002	9,39E-001	7,08E-001	7,37E-001
oxyR	25	4,76E-001	4,81E-001	9,85E-004	5,11E-001	5,58E-002	1,64E-001	6,18E-001	1,72E-006
phoB	30	2,64E-002	6,95E-002	6,87E-001	3,29E-002	7,38E-001	3,59E-003	7,00E-004	2,60E-002
phoP	10	2,54E-002	1,79E-002	9,83E-003	9,51E-001	3,99E-001	8,19E-002	2,55E-001	1,27E-003
purR	46	2,90E-001	1,78E-001	7,23E-001	3,61E-001	6,50E-001	6,50E-001	8,25E-002	4,76E-001
rob	19	2,61E-001	3,30E-001	8,77E-003	1,15E-001	1,67E-001	7,08E-001	3,75E-001	1,52E-003
rpoN	30	6,87E-001	6,33E-001	1,18E-003	3,08E-001	1,42E-001	1,18E-002	1,32E-001	2,10E-003
soxS	38	2,96E-003	4,30E-003	8,92E-007	6,53E-001	4,06E-002	3,49E-003	2,20E-001	1,79E-006
trpR	9	1,63E-002	1,64E-002	7,51E-001	2,83E-001	9,57E-001	4,47E-002	3,38E-003	2,03E-001
tyrR	8	3,93E-001	5,01E-001	1,12E-001	1,82E-001	9,20E-001	5,70E-001	7,11E-001	7,79E-001
yhiX	14	5,75E-004	2,86E-001	2,83E-008	2,00E-008	4,11E-003	9,37E-003	2,42E-005	1,90E-006
yiaJ	9	2,92E-001	3,00E-002	2,15E-001	5,35E-001	1,69E-001	2,26E-002	3,23E-002	6,34E-002

Lebenslauf

Maurice Scheer

Diplom-Biologe
M. Sc. (FH)

wohnhaft in Braunschweig

PERSÖNLICHE ANGABEN

Geburtsdatum:	20.02.1973 in Berlin
Familienstand:	verheiratet mit Nina Hesse-Scheer, ein Kind
Staatsangehörigkeit:	deutsch

SCHULE

1979 – 1983	Südpark-Grundschule in Berlin
1983 – 1992	Gymnasium Canisius-Kolleg in Berlin Schulabschluss: Abitur

WEHRDIENST

1992 – 1993	Wehrpflichtiger bei der Bundesmarine

STUDIUM

1993 – 1999	Biologie-Studium, Freie Universität Berlin **Schwerpunkte**: Molekularbiologie und Biochemie, molekulare Genetik, Immunologie, Tierphysiologie Abschluss: Diplom-Biologe
2003 – 2007	Aufbaustudium Bioinformatik, Technische Fachhochschule Berlin **Schwerpunkte**: Graphentheorie, Petrinetze, bioinformatische Algorithmen, Statistik, Programmieren Abschluss: Master of Science (FH) im postgradualen Studiengang Bioinformatik

BERUFSPRAXIS

10/1999 – 12/1999	Max-Planck-Institut für molekulare Genetik in Berlin Freie Mitarbeit im Fachgebiet molekulare Genetik
02/2000 – 01/2001	Universität Konstanz, Lehrstuhl für Entwicklungsneurobiologie Wissenschaftlicher Mitarbeiter im Fachgebiet molekulare Neurobiologie
04/2001 – 09/2002	Firma Biobase GmbH Braunschweig/Wolfenbüttel Annotation und Entwicklung biologischer Datenbanken, speziell TRANSFAC-Datenbank
10/2002 – 06/2007	Fachbereich Informatik, Fachhochschule Wolfenbüttel sowie Institut für Mikrobiologie, Technische Universität Braunschweig Wissenschaftlicher Angestellter im Fachgebiet Bioinformatik
01/2003 – 06/2007	Durchführung der Dissertationsarbeit „Computational Analysis and Interpretation of Prokaryotic High-throughput Data“ Institut für Mikrobiologie, Technische Universität Braunschweig
seit 08/2007	Institut für Biochemie und Biotechnologie, Technische Universität Braunschweig Wissenschaftlicher Angestellter in der Abteilung Bioinformatik Administration und Weiterentwicklung des Enzyminformationssystems BRENDA, Textmining

Danksagung

Als erstes möchte ich meinem Betreuer Prof. Dr. Dieter Jahn (Leiter des Instituts für Mikrobiologie der TU Braunschweig) ganz herzlich für die Vergabe des spannenden Themas und für seine Unterstützung in allen Phasen dieser Arbeit danken. Seine Motivation, sein Optimismus, die überlassenen Freiräume und sein Vertrauen in mich haben die Erstellung dieser Arbeit enorm gefördert – wenn nicht gar erst möglich gemacht. Außerdem danke ich ihm für die wertvollen Tipps und Diskussionen, die stete Hilfe beim Verfassen von Publikationen sowie die Möglichkeit, an internationalen Workshops und Konferenzen teilnehmen und – neben der Doktorarbeit – bioinformatische Kurse besuchen zu dürfen. Außerdem möchte ich meinem Mentor Prof. Dr. Michael Steinert herzlich für die unkomplizierte und freundliche Übernahme des ersten Referats danken.

Desweiteren möchte ich mich sehr bei Prof. Dr. Frank Klawonn für sein kontinuierliches Interesses an dieser Arbeit bedanken. Seine Fähigkeit, auch komplexe mathematische Sachverhalte verständlich zu erklären, sein Blick für's wesentliche sowie die zahlreichen sehr nützlichen Tipps und fruchtbaren Diskussionen haben mir sehr geholfen, neue Ideen im Bereich der Programmierung, Statistik und Datenauswertung zu entwickeln und effizient sowie erfolgreich umzusetzen. Außerdem danke ich ihm für die Hilfe beim Verfassen von Publikationen und seine Bereitschaft, das Zweitgutachten dieser Dissertation zu übernehmen.

Frau Prof. Dr. Petra Dersch möchte ich für das Interesse an meiner Arbeit und die Übernahme des Vorsitzes der Promotionskommission recht herzlich danken.

Herrn Prof. Dr. Ulrich Klages danke ich vielmals für die Erlaubnis, im Rahmen des Intergenomics-Projektes an der FH Wolfenbüttel, meine Doktorarbeit im Arbeitskreis von Prof. Dr. Jahn anfertigen zu dürfen sowie für das in mich gesetzte Vertrauen und die langjährige Finanzierung, ohne die diese Arbeit nicht möglich gewesen wäre. Außerdem ist sein engagierter Einsatz für die erfolgreich bewilligte Verlängerung des Projektes, welche die Abschlußfinanzierung der Arbeit ermöglichte, hervorzuheben. Darüberhinaus bedanke ich mich für die Möglichkeit, zu internationalen Konferenzen reisen und bioinformatische Kurse besuchen zu dürfen.

Meinem Kollegen Dr. Richard Münch danke ich herzlich für die unzähligen fachlichen, technischen sowie strategischen Tipps, Anregungen und wertvollen Diskussionen, ohne die diese Arbeit sicherlich nicht zu dem geworden wäre, was sie jetzt ist. Zudem danke ich ihm für das in mich gesetzte Vertrauen, die Einarbeitung in *PHP* und die PRODORIC-Datenbank sowie viele Linux- und Latex-spezifische Tipps.

Dr. Karsten Hiller danke ich für die zahlreichen sehr nützlichen Tipps, Tricks und Diskussionen zu *Java*, Web Services, Linux, Datenauswertung und Netzwerk-Technik. Es gab nahezu kein Problem, für das er keine Lösung gewußt hätte. Ebenso möchte ich Boyke Bunk danken für die vielen anregenden technischen Diskussionen und den Austausch von

wertvollen Programmier-Tipps.
Dr. Max Schobert danke ich für die vielen guten Ideen und hilfreichen Diskussionen sowie die erfolgreiche Zusammenarbeit bei der Microarray-Daten-Auswertung. Ihm, Dr. Ida Retter und Dr. Michael Stelzer gebührt außerdem großer Dank für das gründliche Korrekturlesen von großen Teilen dieser Arbeit. Dr. Kerstin Schreiber, Beatrice Benkert, Nelli Bös und Sabrina Thoma danke ich für die Bereitstellung der Microarray-Daten von *Pseudomonas aeruginosa* und die gute Zusammenarbeit bei der Auswertung dieser Daten.
Frau Prof. Dr. Ina Koch (TFH Berlin) danke ich für die fruchtbaren Diskussionen und die wertvollen Tipps beim Verfassen von Publikationsmanuskripten.
Dr. Benedikt Brors (DKFZ Heidelberg) möchte ich für die zahlreichen Tipps und die erfolgreiche Zusammenarbeit beim Preprocessing der Microarray-Expressiondaten mittels Bioconductor danken.
Außerdem danke ich meinen Kollegen aus der Arbeitsgruppe Bioinformatik Susanne Behling, Boyke Bunk, Dr. Claudia Choi, Dr. Andreas Grote, Dr. Karsten Hiller, Isam Haddad, Claudia Hundertmark, Johannes Klein und Dr. Richard Münch für die angenehme Arbeitsatmosphäre, ihre Diskussionbereitschaft und die erfolgreiche Zusammenarbeit.
Dr. Elisabeth Härtig und Dr. Jürgen Moser danke ich für ihre guten Ideen und hilfreichen Diskussionen. Für ihre Kooperationsbereitschaft und den Erfahrungsaustausch im Intergenomics-Projekt danke ich Dr. Lorenz Bülow, Dr. Silke Eckstein, Prof. Dr. Reinhard Hehl, Dr. Ida Retter und Dr. Claudia Täubner. Für die sprachliche Überarbeitung des Manuskripts der JProGO-Veröffentlichung danke ich Frau Privatdozentin Dr. Barbara Schulz.
Dem gesamten Arbeitskreis Jahn sowie allen anderen Arbeitsgruppen des Instituts für Mikrobiologie danke ich für die angenehme Atmosphäre und die stete Hilfsbereitschaft während der Doktorarbeit.
Außerdem möchte ich mich bei meinem neuen Chef, Prof. Dr. Dietmar Schomburg, dafür bedanken, dass er mich zur Fertigstellung der Dissertation und Vorbereitung der Disputation freigestellt hat.
Bei meinen Freunden möchte ich mich für ihre Hilfe und Unterstützung während der Doktorarbeit bedanken. Namentlich seien hier Albert Gevorgyan, Carsten Ihlenburg, Daniel Nitz, Heiko Decker, Marc Zischke, Mario Huster und Sandor Ragaly genannt.
Schließlich möchte ich mich bei meinen Eltern und meiner Schwester sehr für die Unterstützung während der gesamten Doktorarbeit bedanken. Mein letzter und besonders herzlicher Dank richtet sich an meine Frau Nina für ihre immer währende Unterstützung, ihr Verständnis und ihre Motivation sowie an unseren Sohn Bruno für die netten Augenblicke zu dritt.